Санкт–Петербургский государственный университет

СЕЛИВАНОВ Дмитрий Александрович

Выпускная квалификационная работа

Реконструкция первичной вершины взаимодействия в центральном детекторе JUNO с помощью нейронных сетей

Уровень образования: бакалавриат
Направление 01.03.02 «Прикладная математика и информатика»
Основная образовательная программа СВ.5005.2015 «Прикладная математика, фундаментальная информатика и программирование»
Профиль «Исследование и проектирование систем управления и обработки сигналов»

Научные руководители:
доцент, кафедра компьютерного моделирования и многопроцессорных систем,
к.ф. - м.н. Гришкин Валерий Михайлович

младший научный сотрудник,
Лаборатория ядерных проблем ОИЯИ,
Тресков Константин Андреевич

кафедра компьютерного моделирования и многопроцессорных систем,
Якушкин Олег Олегович

Рецензент:
доцент, кафедра теории систем управления электрофизической аппаратурой,
к.ф. - м.н. Козынченко Владимир Александрович

Санкт-Петербург
2019 г.

Содержание

Введение

Стандартная модель (СМ) в настоящее время является наиболее успешной теоретической конструкцией в физике элементарных частиц. Однако, она не является полной, так как не описывает:

- Темную материю и темную энергию;
- Барионную асимметрию Вселенной;
- Гравитационное взаимодействие;
- Нейтринные осцилляции и генерацию масс нейтрино.

Исследования физики нейтрино исторически породили большое количество загадок, таких как дефицит потока солнечных нейтрино, реакторную антинейтринную аномалию, возможные указания на существование добавочных (стерильных) нейтрино, не участвующих в слабом взаимодействии, и других.

Исследования дефицита потока солнечных нейтрино привели к открытию смешивания нейтрино и были отмеченны Нобелевской премией в 2015 году. Другие аномалии также могут указывать на пробел в Стандартной модели, привести к экспериментальному открытию физики за пределами СМ и ограничить многочисленные существующие модели физики вне СМ.

Точное определение параметров смешивания и осцилляций нейтрино является важной задачей для проверки Стандартной Модели. Будущие эксперименты, такие как JUNO, DUNE и Hyper-Kamiokande обеспечат измерения фазы нарушения CP-инвариантности и параметров осцилляций с беспрецедентной точностью, а также смогут определить иерархию масс нейтрино, что приведет к долгожданной эре точных измерений в физике нейтрино.

JUNO Одним из будущих экспериментов, связанных с изучением нейтрино, является *Jiangmen Underground Neutrino Observatory* (JUNO) [1].

Этот эксперимент спроектирован для определения иерархии масс нейтрино и измерения трех параметров осцилляции нейтрино с точностью лучше процента, используя поток антинейтрино от атомных электростанций Янцзан и Тайшан.

Детектор будет располагаться рядом с городом Кайпин, префектура Цзянмынь, провинция Гуандун на глубине 700 метров под землей на оптимальном расстоянии примерно в 53 километра от атомных электростанций.

Центральный детектор JUNO – это сферический детектор, наполненный 20-ю килотоннами жидкого сцинтиллятора. Упрощенная схема центрального детектора изображена на рис. 1. Его радиус составляет примерно 19.5 метров. Такой детектор чувствителен к электронным антинейтрино посредством реакции обратного бета-распада $\bar{\nu}_e + p \to e^+ + n$. Поверхность акриловой сферы покрыта около 18000 фотоэлектронными умножителями (ФЭУ), каждый из которых имеет размер примерно 51 сантиметр в диаметре. ФЭУ расположены в порядке шестиугольной сетки, покрывая 78 % поверхности сферы. В поверхности детектора существуют пробелы, созданные для подключения электрического оборудования, а также отверстие с верхней стороны для установки калибровочного оборудования. В небольшом пространстве между большими ФЭУ расположены дополнительные маленькие ФЭУ с диаметром примерно в 3 дюйма. Они используются для снижения систематических неопределенностей, связанных с энергетическим откликом детектора, и более точной регистрации времени информацию о времени прихода фотонов.

Такой детектор открывает возможности для широкой физической программы[1]:

- Определение иерархии нейтрино на уровне статистической значимости в 3—4σ;
- Измерение угла смешивания θ_{12} и расщеплений масс Δm^2_{21} и Δm^2_{32} с точностью менее процента;
- Сбор рекордной статистики антинейтрино от распадов радиоактивных изотопов в земной коре и мантии;

- Измерение диффузного потока антинейтрино от сверхновых;
- Поиск распада протона, предсказываемого разными моделями физики вне Стандартной модели;
- Измерения потоков и спектров солнечных и атмосферных нейтрино;
- Поиски нестандартных взаимодействий нейтрино.

Статистическая значимость верного определения иерархии масс нейтрино в эксперименте сильно зависит от точности реконструкции энергии (энергетического разрешения) — точность в 3% на 1 МэВ соответствует 3σ статистической значимости. Точность реконструкции энергии зависит от положения события в детекторе из-за поглощения, перерассеяния и отражения света от поверхности акриловой сферы, что влияет на точность определения иерархии масс, как показано в [3]. Таким образом для улучшения энергетического разрешения необходимо реконструировать пространственное положения вершины взаимодействия в детекторе с высокой точностью.

Также точное восстановления положения области энерговыделения в сцинтилляторе необходимо для:

- Отбора событий обратного бета-распада. Одним из условия отбора является пространственное расстояния не больше 1.5 м. между пер-

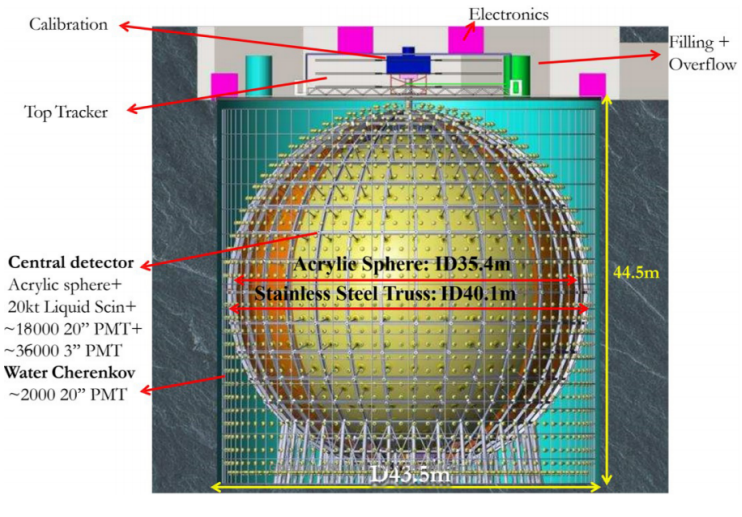


Рис. 1: Упрощенная схема центрального детектора JUNO [2].

вичным сигналом от позитрона и вторичном от захвата нейтрона на водороде;

- Отсечения фоновых событий, производимых во взаимодействиях атмосферных мюонов с веществом детектора. Вокруг каждого мюонного трека будет вырезаться область пространства во временном окне примерно в 1 сек. и все события, реконструированные в данной области будут отбрасываться;

- Выделения доверительного объёма в детекторе для отсечения событий из области около поверхности акриловой сферы для снижения количества фоновых событий от внешних радиоактивных источников.

Таким образом для выполнения физической программы эксперимента необходимы надёжные и точные алгоритмы восстановления энергии и положения вершины взаимодействия в центральном детекторе JUNO. Также из-за большого количества данных (примерно 2 ПБ/год), которые будут поступать в процессе эксперимента, необходимо чтобы эти алгоритмы были вычислительно производительными.

Постановка задачи

Для каждого из событий взаимодействия позитрона e^+ с веществом детектора известны:

- Координаты расположения всех 18000 больших ФЭУ;

- Суммарное количество фотонов, захваченных каждым ФЭУ;

- Времена срабатывания каждого ФЭУ;

- Идентификатор того, что срабатывание было шумовым.

Задача состоит в разработке метода на основе нейронных сетей, который по вышеописанным данным для каждого из событий будет определять координаты вершины взаимодействия, то есть координаты столкновения позитрона с электроном. Для оценки точности реконструкции этой

величины используется значение разницы реконструированного и истинного радиусов точки реакции:

$$\Delta R = R_{rec} - R_{true}$$

Или, что то же самое:

$$\Delta R = \sqrt{x_{rec}^2 + y_{rec}^2 + z_{rec}^2} - \sqrt{x_{true}^2 + y_{true}^2 + z_{true}^2}$$

При этом требуется, чтобы значение стандартного отклонения $\sigma(\Delta R)$

$$\begin{aligned} &\sigma(\Delta R) = \sqrt{\frac{\sum_{i=1}^{n}(\Delta R_i - \Delta\bar{R})}{n}} \quad , \\ &\Delta R_i = R_i^{rec} - R_i^{true} \quad , \\ &\Delta\bar{R} = \frac{\sum_{i=1}^{n}\Delta R_i}{n} \end{aligned}$$

было меньше или равно 7 см для событий с энергией 1 МэВ. Именно такой точностью обладает классический метод наибольшего правдоподобия в JUNO [4].

Визуализация геометрических путей, которые прошли в детекторе фотоны, образованные после взаимодействия позитрона с веществом детектора представлена на рис. 2. В упрощенной формулировке задача состоит в том, чтобы разработать метод с использованием нейронных сетей, способный определить геометрическую точку в пространстве, из которой были выпущены эти фотоны.

Обзор литературы

При подготовке данной работы для ознакомления с общими задачи проекта JUNO были использованы статьи [1], [2], [6] и [7], . Для получения представления о процессе реконструкции первичной вершины взаимодействия с помощью классического метода максимального правдоподобия была использована публикация [4]. Указанные источники содержат исчерпывающую информацию об экспериментах, которые планируется проводить

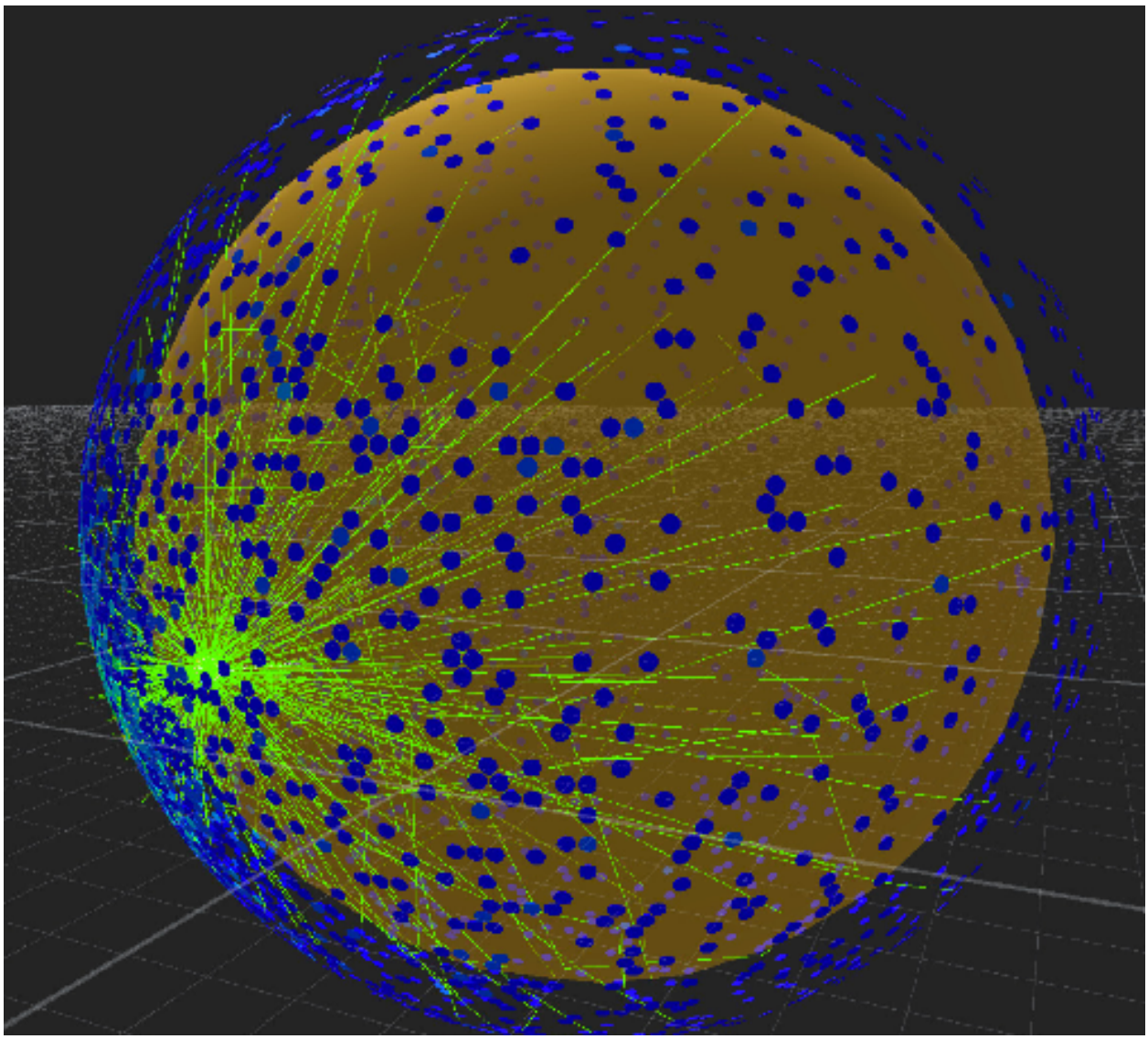


Рис. 2: Визуализация геометрических путей фотонов, образованных в вершине взаимодействия и зарегистрированных с помощью ФЭУ. Синие точки обозначают фотоэлектронные умножители, которые зарегистрировали хотя бы один фотон, а зеленые линии – геометрические пути, которые прошли фотоны [5].

в рамках проекта JUNO. Также здесь можно найти описание самого центрального детектора, информацию о принципе детектирования нейтрино в жидкосцинтилляторных детекторах, параметрах осцилляции нейтрино и фундаментальных проблемах в физике нейтрино. Кроме этого, статья [5] предоставляет наглядную визуализированную информацию о структуре детектора и о физических событиях в детекторе.

Книга [8] содержит описание используемых в данной работе методов машинного обучения, а также математическое описание процесса обучения нейронных сетей. Здесь можно найти подробное описание архитектуры

многослойного перцептрона и сверточных нейронных сетей. В ней также приведены недостатки и проблемы, которые могут возникнуть при использовании нейронных сетей для анализа данных, а также рекомендации по способам минимизации их влияния.

Публикация [9] содержит информацию об остаточных блоках, которые используются в остаточных сверточных сетях, а также причины их использования. Публикации [10] и [11] содержат описание методов регуляризации, примененных в данной работе.

Для получения сравнительной информации о способах проекции сферы на плоскость была использована статья [12]. Метод проекции сферической структуры детектора на плоскость используется для генерации входных данных в сверточной и остаточной сверточной нейронных сетях.

Немаловажной публикацией является [13], которая представляет подробную информацию о программной библиотеке Tensorflow, использованной в данной работе для практической реализации нейронных сетей. Эта библиотека поддерживает вычисления на видеопроцессорах (GPU), а также описывает информацию о графовой структуре вычислений в Tensorflow.

Глава 1. Классические методы: метод среднего заряда, метод наибольшего правдоподобия

Реконструкция вершины взаимодействия событий внутри жидкосцинтилляторного детектора обычно производится с помощью математических методов и алгоритмов приближения. В данной главе будет рассматриваться краткий обзор точности и эффективности существующих методов.

1.1 Метод среднего заряда

Для сферических детекторов наиболее простым и интуитивным способом определить координаты вершины взаимодействия является метод среднего заряда. Геометрическая точка, соответствующая центру заряда события внутри детектора определяется как:

$$\bar{\boldsymbol{x}} = \frac{\sum_i \boldsymbol{x} \cdot n^i_{\text{p.e.}}}{\sum_i n^i_{\text{p.e.}}},$$

где $\boldsymbol{x}$ — это позиция i-го ФЭУ и $n^i_{\text{p.e.}}$— это количество фотонов, зарегистрированных i-ым ФЭУ в этом событии.

В качестве примера, 156 тысяч симулированных (с помощью метода Монте-Карло) событий взаимодействия позитрона с веществом детектора были проанализированы этим методом. Энергия позитронов в данном примере составляет 5 МэВ, координаты вершин взаимодействия равномерно распределены внутри детектора. Результаты показаны на рисунке 3. Данный рисунок показывает, что стандартное отклонение и систематическое смещение определения вершины взаимодействия сильно возрастают при возрастании истинного радиуса вершины. В особенности, когда событие происходит вблизи края детектора ($r > 16$ м).

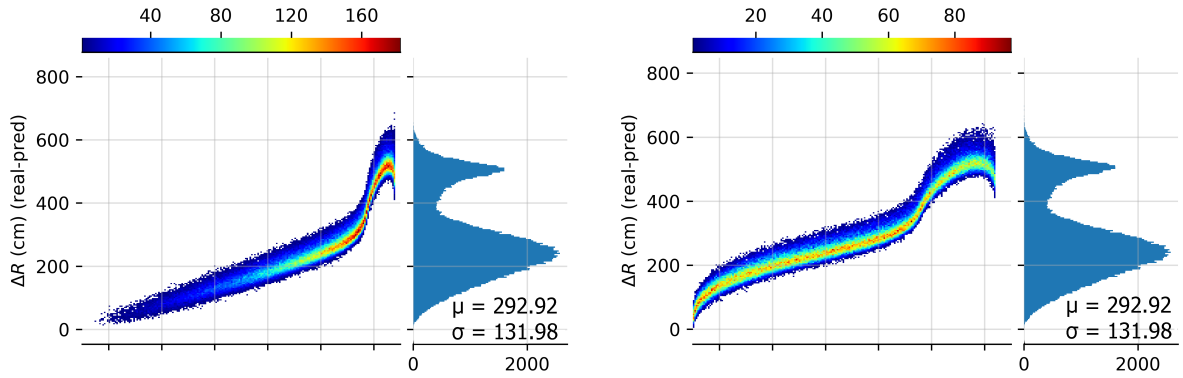


Рис. 3: Результаты реконструкции вершины взаимодействия с помощью метода среднего заряда на 156 тысячах событий с энергией 5 МэВ. На графике показана разница в евклидовом расстоянии между истинным и реконструированными положениями вершины взаимодействия, построенная в зависимости от радиуса (слева) и радиуса в кубе (справа).

Результаты по отдельным координатам показаны на рисунке 4. В данном случае, ошибка в определении радиуса события вносит наибольший вклад. Ошибка в определении сферического угла ϕ стабильна практически при всех значениях радиуса вершины и ее стандартное отклонение составляет примерно ≈ 0.04 радиан. Ошибка в определении угла θ немного изменяется в зависимости от радиуса. Возможной причиной этого является асимметричность расположения ФЭУ в детекторе. Стандартное отклоне-

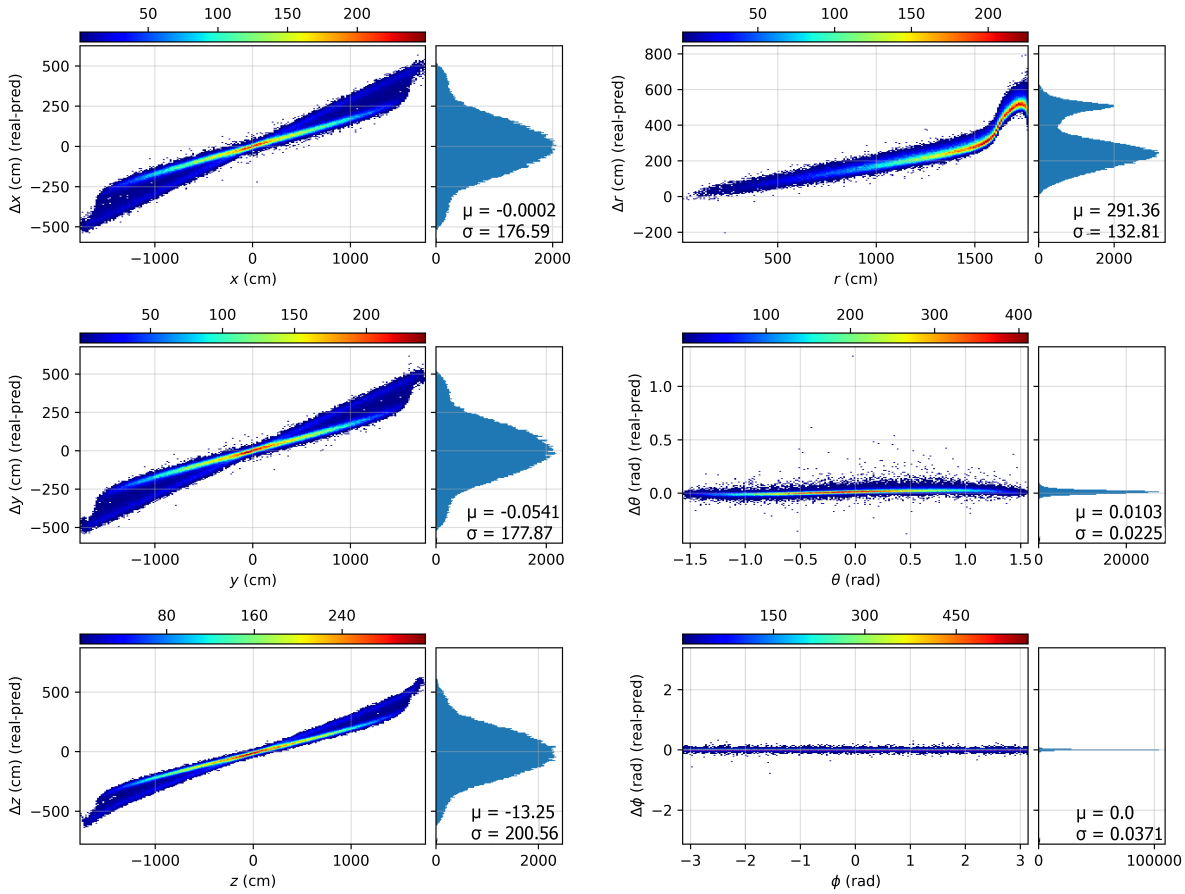


Рис. 4: Результаты реконструкции вершины взаимодействия с помощью метода среднего заряда на 156 тысячах событий с энергией 5 МэВ. На графике показана разница в евклидовых и сферических координатах между истинным и реконструированным значениями вершины взаимодействия.

ние этой величины составляет приблизительно 0.023 радиан.

Абсолютное значение расстояния между реконструированным и истинным значениями радиусов вершины взаимодействия возрастает до 6 метров, когда событие располагается близко к краю детектора. Это вызвано полным внутренним отражением фотонов от поверхности акриловой сферы и их детектированием в более отдаленных ФЭУ. Причина наклона условных кривых на рисунке объясняется в [4] и может быть скорректирована с помощью коэффициента $\approx \frac{6}{5}$.

1.2 Метод максимального правдоподобия

На данный момент наиболее точным классическим методом реконструкции вершины взаимодействия в центральном детекторе JUNO явля-

ется метод максимального правдоподобия. Этот метод использует функцию правдоподобия для света дойти до ФЭУ за наблюдаемое время при заданной вершине взаимодействия. Модель распространения света учитывает зависимость скорости света в жидком сцинтилляторе от длины волны, вероятности поглощения, отражения, перерассеяния и регистрации света данным фотоумножителем и временное распределение начальной вспышки. Модель предсказывает среднее время срабатывания каждого ФЭУ вместе с вероятностью ненаблюдения сигнала при его отсутствии. Положение вершины определяется путем максимизации функции правдоподобия.

Данный алгоритм реконструкции способен определить координаты вершины взаимодействия со стандартным отклонением в 7 сантиметров для событий с энергией в 1 МэВ [4].

Основным недостатком метода максимального правдоподобия является то, что он требует больших вычислительных затрат. Реконструкция вершины для одного события может занимать десятки секунд, в особенности для событий с высокими энергиями. Нейронные сети могут быть решением данной проблемы, так как они требуют на порядок меньше времени для анализа данных и работают за постоянное время после того, как они успешно обучены.

Глава 2. Теория

В данной работе перевод терминов из теории нейронных сетей с английского языка на русский соответствует переводу аналогичных терминов в [8].

2.1 Нейронные сети

Машинное обучение — это набор алгоритмов, использующихся для решения различных и сложных задач. Нейронные сети являются большой частью машинного обучения. Основной целью нейронных сетей является аппроксимация некоторой функции $y = f(x, \theta)$, которая является отображением вектора входных значений x в вектор выходных значений y. В процессе обучения происходит изменение значений матрицы параметров θ

с помощью минимизации значения *функции потерь* (*loss function*) таким образом, чтобы нейронная сеть выдавала выходные данные наиболее близкие к правильным для заданных входных данных. Нейронные сети называются *сетями*, потому что они состоят из слоев, в каждом из которых есть нейроны. Мы можем понимать нейронные сети как некоторую комбинацию функций, которые применяются последовательно:

$$f(\cdot) = f_3(f_2(f_1(\cdot)))$$

В таком случае, f_1 обозначает первый слой, f_2 второй и так далее. Наиболее важные аспекты нейронных сетей могут быть показаны на примере *многослойного перцептрона* (иначе *полносвязной сети прямого распространения* или *fully connected feedforward network*) (рис.5).

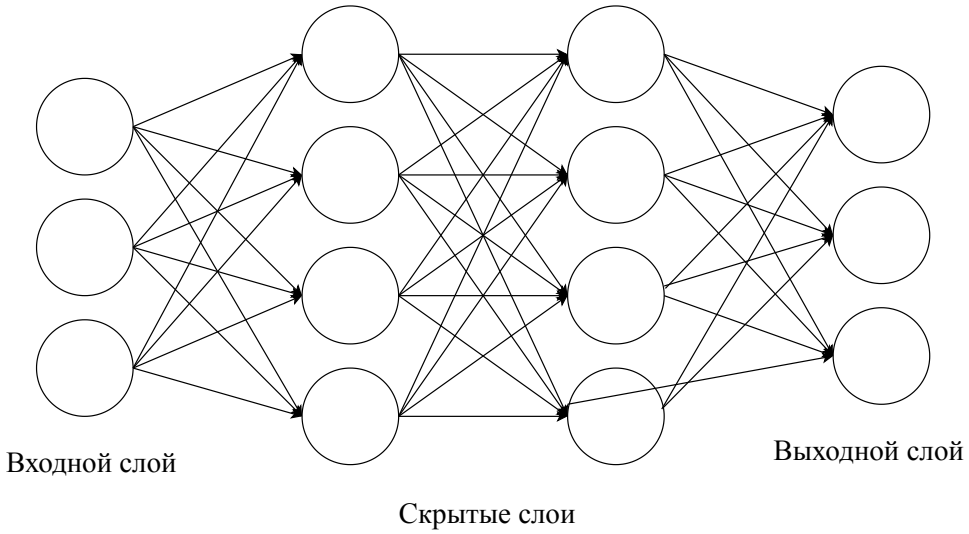


Рис. 5: Простой пример архитектуры многослойного перцептрона. Все слои между входным и выходным слоями называются *скрытыми* (*hidden layers*), поскольку набор данных для обучения не содержит желаемых выходных данных для этих слоев. Каждый нейрон в таких слоях называется *скрытым*.

Выходное значение i-го нейрона в k-ом слое может быть найдено согласно следующему выражению:

$$y_i^{(k)} = f_i^{(k)} \left(\sum_j w_{ij}^{(k)} \cdot x_j + b_i^{(k)} \right),$$

где $f_i^{(k)}$ — функция активации этого нейрона, $w_{ij}^{(k)}$ — *весовой коэффициент*

нейрона по отношению к j-ому входному значению, x_j — j-ое входное значение (если k — номер следующего слоя после входного) или j-ое выходное значение $(k-1)$-го слоя (если k — это номер остальных слоев) и $b_i^{(k)}$ — это *порог* (*bias* или *threshold*) этого нейрона.

В этом случае набор θ параметров сети состоит из весовых коэффициентов w_i и порогов b_i. Наиболее широко распространенным алгоритмом для минимизации функции потерь (нахождения лучшего набора параметров θ) является *градиентный спуск* (*gradient descent*). Однако, в машинном обучении применяется много таких алгоритмов и какой из них выбрать — зависит от конкретной задачи. Глубокие сети имеют огромное количество обучаемых параметров, так что требуется выбрать наиболее эффективный метод для вычисления градиентов. Для этого широко используется *метод обратного распространения ошибки* (*backpropagation algorithm*) [8].

Для того, чтобы задать нейронную сеть, необходимо выбрать:

- Архитектуру (тип/количество слоев, количество нейронов в каждом слое);
- Функции активации для каждого слоя;
- Гиперпараметры;
- Функцию потерь;
- Метод минимизации функции потерь.

Функция потерь ставит в соответствие разнице между предсказанными и истинными выходными значениями вещественное число. Мы можем выбрать функцию потерь так, чтобы она имела интуитивный физический смысл, но на практике используются функции, которые имеют меньшее количество операций вычисления. К примеру, в задачах регрессии часто используется *среднеквадратическая ошибка* (mean squared error):

$$\mathrm{MSE} = \frac{1}{n}\sum_i \left(y_i^{\mathrm{true}} - y_i^{\mathrm{pred}}\right)^2,$$

где y^{true} — это желаемый выходной вектор, состоящий из n компонент, и y^{pred} — это предсказанный нейронной сетью выходной вектор из n компонент.

Функция активации (*activation function*) нейрона определяет выходное значение этого нейрона в соответствии со входными значениями. Только нелинейные функции позволяют нейронным сетям решать сложные задачи. Обычно, функция активации в нейронных сетях определяется одинаково для всех нейронов в слое.

Очень популярной функцией активации является *функция Ферми* или *сигмоидальная функция* (рис. 6):

$$f(y) = \frac{1}{1 + e^{-y}},$$

которая отображает диапазон входных значений $(-\infty; +\infty)$ на интервал $(0; 1)$. Также часто используемой функцией является *гиперболический тангенс* (рис. 6), который отображает диапазон входных значений $(-\infty; +\infty)$ на интервал $(-1; 1)$.

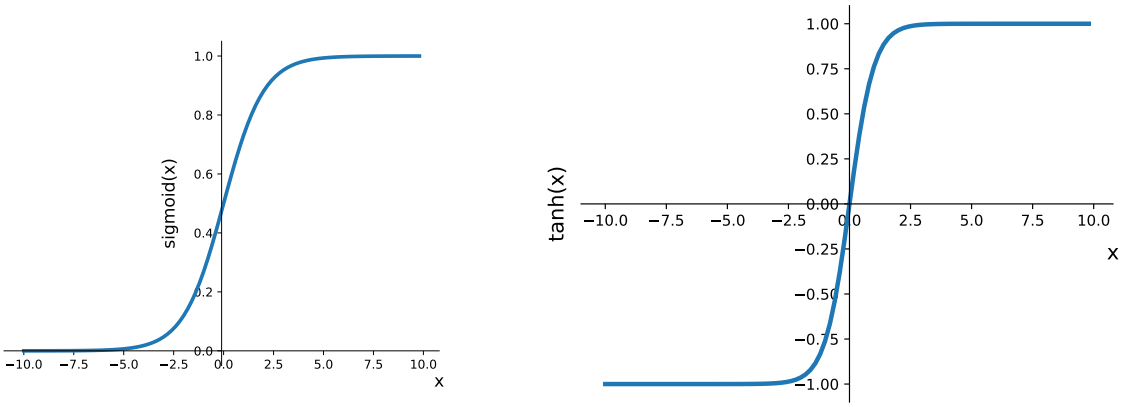


Рис. 6: Сигмоидальная функция (слева) и гиперболический тангенс (справа).

Скорость обучения — это параметр, который определяет величину изменения весовых коэффициентов по отношению к градиенту функции потерь. Он обозначает длину шага обучения. Математически, это означает

следующее:

$$\theta_{n+1} = \theta_n - \alpha \frac{\partial}{\partial \theta_n} J(\theta_n),$$

где α — это скорость обучения, $J(\theta_n)$ — значение функции потерь.

Гиперпараметры (*hyperparameters*) — это переменные, которые определяют процесс обучения и архитектуру сети. Эти параметры задаются до начала обучения. Например, *скорость обучения* (*learning rate*) в градиентном спуске, *коэффициент регуляризации* (см. 2.3), количество эпох обучения, размер одного *пакета обучения* (*batch size*) и так далее. Также, количество нейронов и количество слоев в сети могут тоже рассматриваться как гиперпараметры, когда мы хотим найти наиболее подходящую архитектуру для решения конкретной задачи.

Эпоха — это итерация обучения, которая включает использование всех примеров из обучающей выборки по 1 разу. Обучение нейронной сети в количестве n эпох означает, что каждый пример из обучающей выборки был использован n раз.

К сожалению, если размерность вектора входных значений сети большая — это приводит к огромному количеству обучающихся параметров (θ). К примеру, если мы имеем информацию о заряде каждого из 20000 ФЭУ для конкретного события в детекторе, 2 скрытых полносвязных слоя с 20000 нейронов в каждом, 1 выходной слой с 3 нейронами, тогда количество только весовых коэффициентов в полносвязной сети прямого распространения будет:

$$20\,000^2 + 20\,000^2 + 20\,000 \cdot 3 = 800\,060\,000$$

Именно поэтому, наиболее разумно использовать другие типы нейронных сетей, которые могут использовать большое количество входных значений, получать из них наиболее важную информацию, используя меньшее количество обучающихся параметров.

2.2 Сверточные нейронные сети

Обычные нейронные сети преобразуют входные значения, пропуская их через некоторое количество скрытых слоев, чтобы получить желаемые выходные значения. *Сверточные нейронные сети* (*convolutional neural network*) немного отличаются. Во-первых, они позволяют специализировать сети для работы с данными, имеющими четко выраженную сеточную топологию (например, с матрицами), и хорошо масштабировать такие модели к задачам очень большого размера. Особенно успешным этот подход оказался в применении к двумерным изображениям. К примеру, компьютерное изображение может быть рассмотрено как 2D матрица, в каждой ячейке которой стоит значение соответствующего пикселя. Во-вторых, сверточные нейронные сети имеют два основных компонента:

- Часть выделения карты признаков (*feature extraction part*), которая ответственна за получение наиболее важной информации и уменьшение размерности данных;
- Часть, которая непосредственно решает задачу регрессии/классификации на основе информации, полученной из предыдущего компонента сети.

В первой части сеть производит операции *свертки* (*convolution operation*) и операции *пулинга* (*pooling operation*), чтобы получить наиболее важную информацию из входных значений. К примеру, если изображение содержит человеческое лицо — тогда нейронная сеть *может* распознать некоторые части лица — уши, глаза, губы и так далее.

Во второй части нейронная сеть использует вышеописанную информацию, чтобы произвести предсказания и получить выходные значения, пропуская эту информацию через набор скрытых слоев. Архитектура этого компонента идентична полносвязной сети прямого распространения.

Операция свертки (рис. 7) — одна из основных блоков любой сверточной нейронной сети. Она производится с помощью *фильтра*, который математически является матрицей весовых коэффициентов. Фильтр перемещается по входному изображению (или матрице). В каждой позиции фильтра

производится поэлементное умножение матрицы весовых коэффициентов (фильтра) и соответствующей части матрицы входных значений, а затем производится суммирование всех элементов полученной матрицы, результат которого формирует новый пиксель на результирующем изображении — *карте признаков* (*feature map*). Обычно, каждый сверточный слой имеет набор из n фильтров, которые производят n карт признаков.

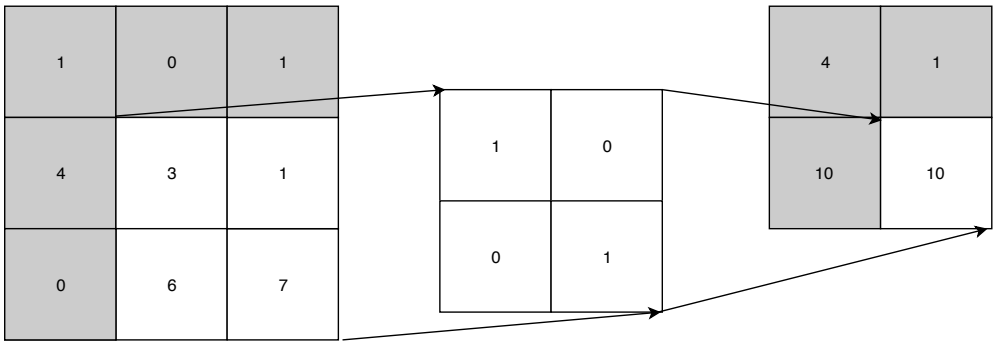


Рис. 7: Визуализация операции свертки.

После операции свертки применяется функция активации. В случае сверточной нейронной сети, это часто *блок линейной ректификации* (*Rectified Linear Unit* или *ReLU*) (см. рис. 8):

$$f(x) = \max(0, x)$$

Согласно [14], сверточные нейронные сети с очень глубокими архитектурами обучаются в несколько раз быстрее, если используют функцию активации ReLU вместо гиперболического тангенса.

Частой практикой является использование слоя *пулинга* (*pooling*, иначе слой *субдискретизации* или слой *подвыборки*) после одного или нескольких слоев свертки. Пулинг так же, как и свертка, имеет фильтр конкретного размера, который перемещается по изображению и производит операции *максимального* (*max pooling*) [15] или *усредняющего* (*average pooling*) пулинга. Эти операции сильно уменьшают размерность данных (см. рис. 9).

В процессе стадии предсказывания каждый параметр многослойного перцептрона используется только один раз. В случае сверточной нейронной сети, каждый элемент матрицы весовых коэффициентов фильтра ис-

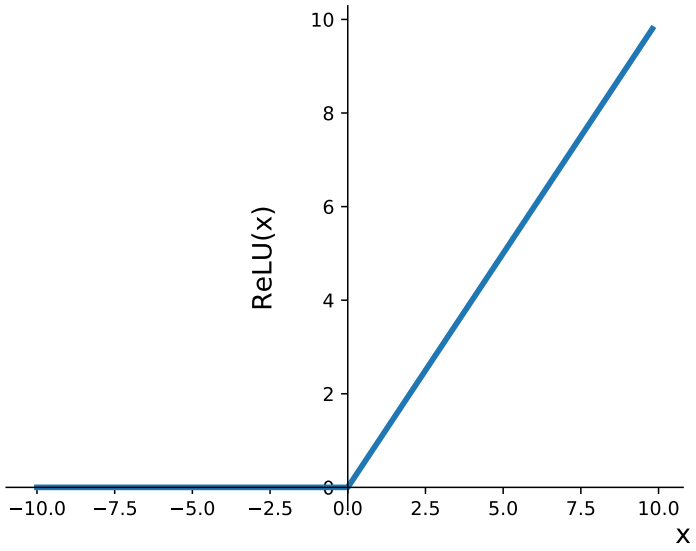


Рис. 8: График функции активации ReLU.

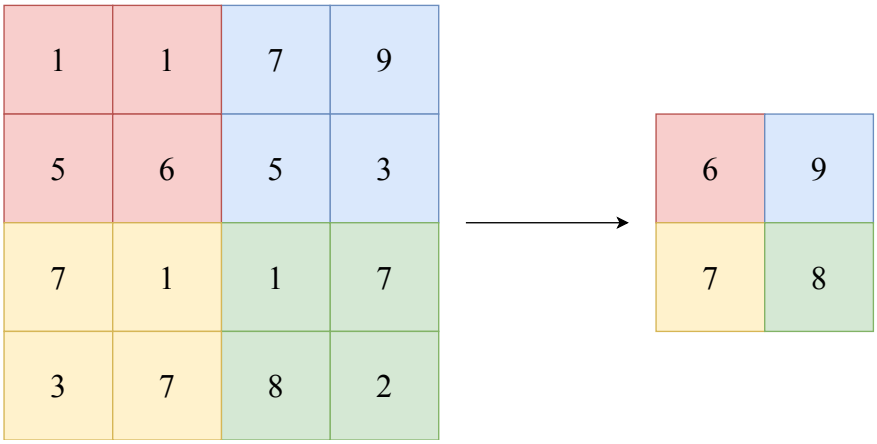


Рис. 9: Пример операции максимального пулинга на матрице размера 4x4.

пользуется при вычислениях на каждом входном пикселе. В комбинации с использованием слоев пулинга, которые уменьшают размерность данных, которые в итоге поступят на вход скрытым полносвязным слоям — это очень сильно снижает количество весовых коэффициентов (т.е. обучаемых параметров).

Двумя основными проблемами обучения глубоких сетей являются проблема взрывных градиентов [16] и проблема затухающих градиентов [17]. При дифференцировании по цепному правилу до самых глубоких слоев нейронной сети доходит очень маленькая величина градиента. Против

этой проблемы помогает бороться так называемый *остаточный блок* (*residual block*) [9]. В таком случае, для пары некоторых слоев добавляется дополнительная связь, которая проходит мимо них. Пусть z^k — это вектор выходных значений k-го слоя до применения функции активации, а a^k — выход после. Тогда математически остаточный блок означает следующую операцию:

$$a^{k+2} = f(z^{k+2} + a^k),$$

где f — функция активации. Такая нейронная сеть учится предсказывать функцию $\mathcal{F}(x) - x$ вместо $\mathcal{F}(x)$. Для компенсации этой разницы вводится замыкающее соединение (shortcut connection), которое добавляет недостающий x к функции. Предположение авторов в [9] заключается в том, что такую разностную функцию проще обучать, что помогает бороться с проблемой деградации нейронной сети при увеличении ее глубины. Изображение структуры остаточного блока представлено на рис. 10.

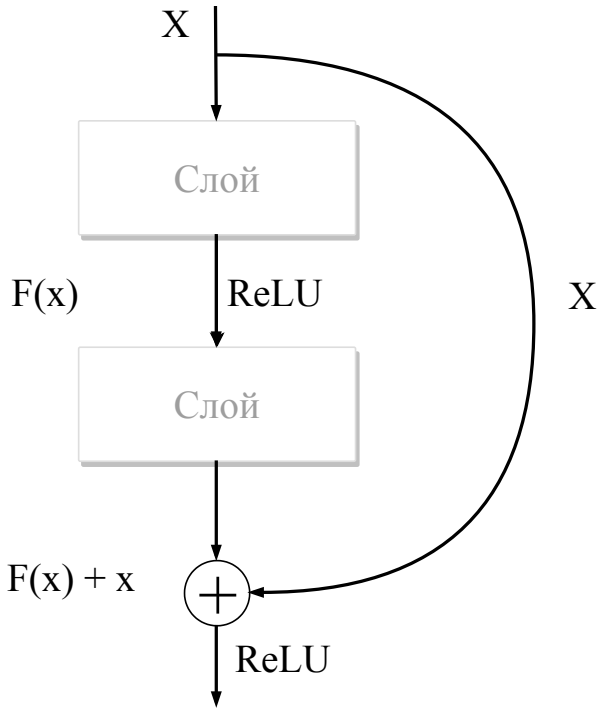


Рис. 10: Визуализация одной из возможных архитектур остаточного блока.

2.3 Методы регуляризации

В процессе обучения ожидается, что значение функции потерь для обучающей и тестовой выборок будет уменьшаться из-за изменения весовых коэффициентов, которое происходит на каждом шаге обучения в процессе работы метода минимизации функции потерь. На рисунке 11 изображен пример, в котором значение функции потерь на обучающей выборке снижается, но в некоторый момент значение функции потерь на тестовой выборке начинает расти. В машинном обучении это называется *переобучением* (*overfitting*). Переобучение — негативное явление, возникающее, когда алгоритм обучения вырабатывает предсказания, которые слишком близко или точно соответствуют конкретному набору данных и поэтому не подходят для применения алгоритма к дополнительным данным или будущим наблюдениям. Чтобы избежать этого, используется ряд методов регуляризации.

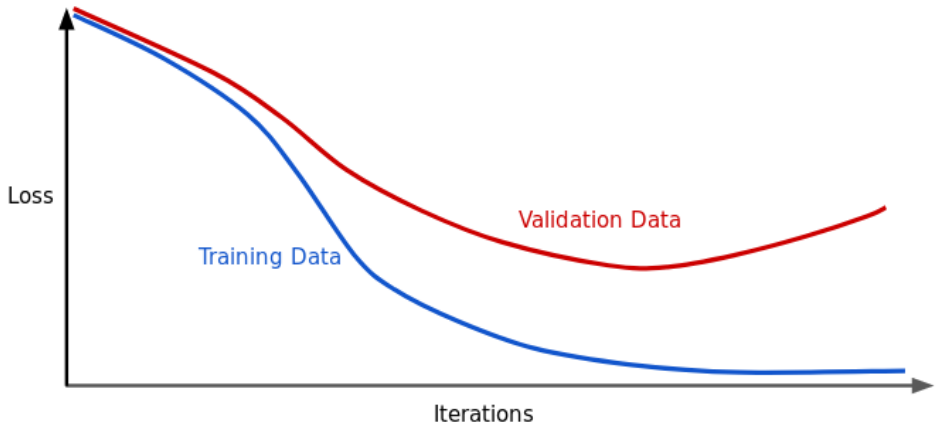


Рис. 11: Пример эффекта переобучения.

2.3.1 Прореживание

Слой *прореживания* (*dropout*) состоит в том, что выходное значение случайного нейрона/нейронов заменяется на 0 при каждом обновлении параметров обучения. Каждый нейрон имеет вероятность p остаться нетронутым и вероятность $(1 - p)$ быть отброшенным (его выходное значение станет равно 0). Выходные значения нейронов, которые не были отброшены, умножаются на коэффициент $\frac{1}{1-p}$, чтобы их сумма не изменялась в

процессе обучения и в процессе предсказания. Наиболее частой практикой является применение прореживания только в процессе обучения, то есть все нейроны остаются нетронутыми на стадии тестирования / предсказания.

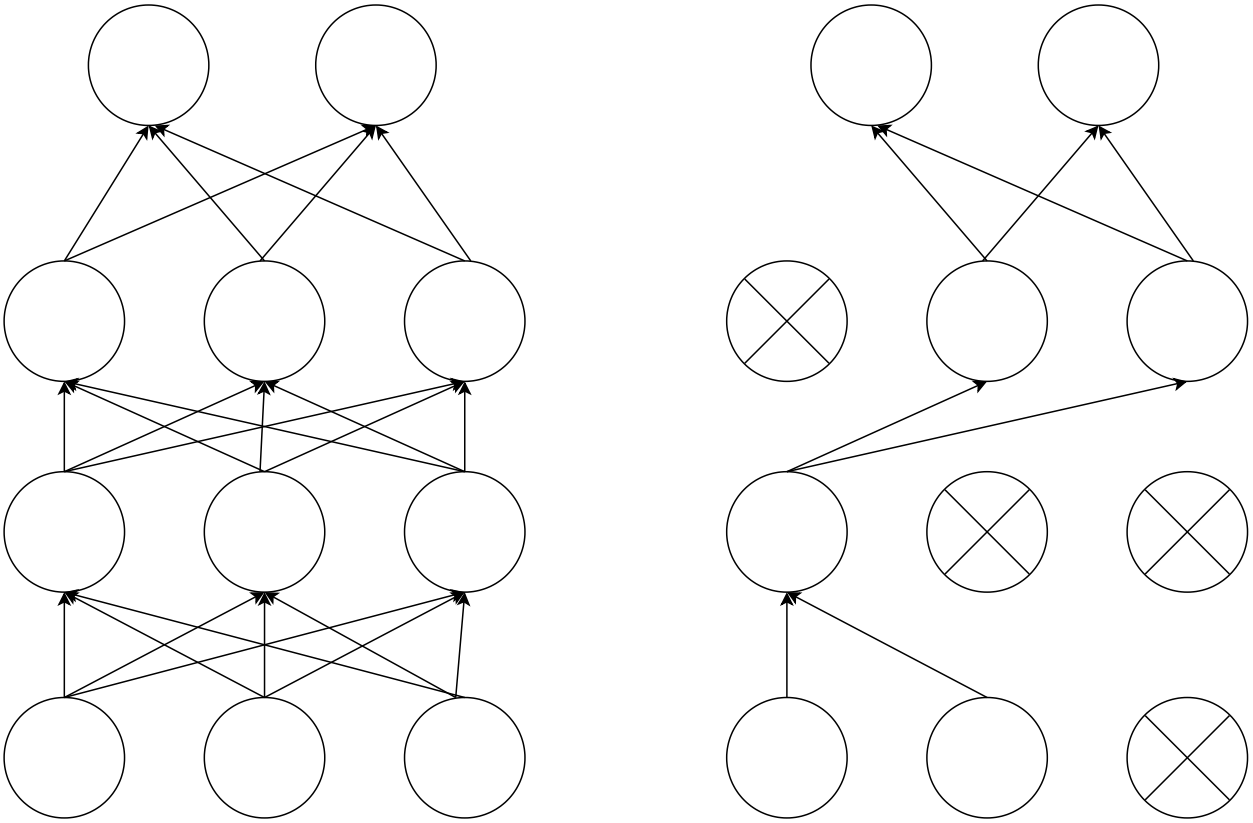


Рис. 12: Операция прореживания.

Схематическое представление операции прореживания изображено на рис. 12. Согласно [10], операция прореживания позволяет улучшить регуляризацию сети и заставляет слои замечать схожие “признаки” разными нейронами независимо.

2.3.2 Пакетная нормировка

Пакетная нормировка (*batch normalization*) [11] — это метод адаптивной репараметризации, появившийся из-за трудностей обучения очень глубоких моделей. Он состоит в замене выходных значений H некоторого слоя на $\gamma H' + \beta$, где γ и β — обучающиеся параметры, а H' определяется

следующим образом:

$$H' = \frac{H - \mu}{\sigma},$$
$$\mu = \frac{1}{n}\sum_{i=1}^{n} H_i,$$
$$\sigma = \sqrt{\epsilon + \frac{1}{n}\sum_{i=1}^{n}(H - \mu)_i^2},$$

где ϵ — небольшое положительное число, например 10^{-8}, введенное, чтобы избежать неопределенного градиента $\sqrt{z}$ в точке $z = 0$. Практически, пакетная нормировка позволяет каждому слою обучаться чуть более независимо от других слоев, что улучшает регуляризующую способность сети.

2.3.3 L2 регуляризация

Вместо того, чтобы минимизировать значение функции потерь:

$$\theta = \arg\min_{\theta}(\text{loss}(x, \theta)),$$

мы будем минимизировать сумму сложности модели и значения функции потерь:

$$\theta = \arg\min_{\theta}(\text{loss}(x, \theta) + \text{complexity}(\theta))$$

В терминах L2 регуляризации, сложность модели определяется как:

$$\text{complexity}(\theta) = \frac{\alpha}{2} \cdot \|w\|_2^2 = \frac{\alpha}{2} \cdot (w_1^2 + w_2^2 + ...),$$

где α — это коэффициент регуляризации в диапазоне $[0; 1]$ и w — вектор всех весовых коэффициентов сети.

Регуляризация накладывает штраф на очень большие значения весовых коэффициентов. Это вынуждает весовые коэффициенты стремиться к меньшим значениям. Этот метод помогает бороться с переобучением. Коэффициент регуляризации часто выбирается эмпирически и может рассматриваться как один из гиперпараметров нейронной сети.

Глава 3. Проекция

Множество методов проекции поверхности сферы на плоскость были изобретены в прошлом. Обзор методов проекции и их особенностей может быть найден в [18].

В этой работе были использованы две проекции: проекция Мольвейде [19] и синусоидальная проекция [20]. Проекция Мольвейде осуществляется с помощью следующих выражений:

$$x = 2\sqrt{2}R\frac{\lambda}{\pi}\cos\theta,$$
$$y = \sqrt{2}R\sin\theta,$$

где θ — это вспомогательный угол, определяемый выражением:

$$\pi\sin\phi = 2\theta + \sin 2\theta$$

и λ — это долгота, ϕ — широта и R — радиус. Так как координаты ФЭУ даны в Евлидовых координатах, то они должны быть преобразованы в сферические согласно стандартным выражениям. Пример проекции поверхности Земли с помощью проекции Мольвейде изображен на рис. 13.

Рис. 13: Пример проекции поверхности Земли с помощью проекции Мольвейде [19].

Синусоидальная проекция определяется следующими выражениями:

$$x = \lambda \cos \phi,$$
$$y = \phi$$

Масштаб проекции в направлении север-юг одинаков на центральном меридиане. В направлении запад-восток масштаб соответствует реальному, длина каждой параллели пропорциональна косинусу широты, поэтому карта ограничена справа и слева двумя повернутыми ветвями косинусоиды. Пример проекции поверхности Земли с помощью синусоидальной проекции изображен на рис. 14.

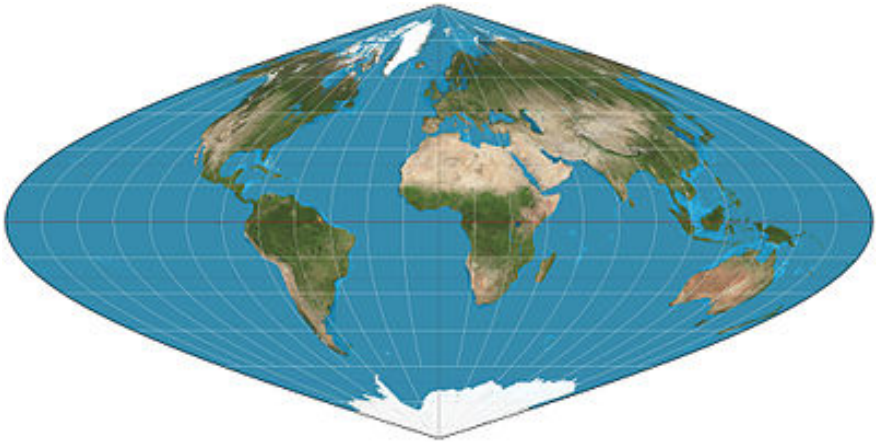


Рис. 14: Пример проекции поверхности Земли с помощью синусоидальной проекции [20].

Глава 4. Реконструкция первичной вершины взаимодействия

В данной главе рассмотрены три метода реконструкции вершины взаимодействия в центральном детекторе JUNO:

- С использованием многослойного перцептрона;
- С использованием сверточной нейронной сети;
- С использованием остаточной сверточной нейронной сети.

Эти методы были разработаны и протестированы в рамках данной работы. Для каждого из методов описываются входные данные, архитектура и результаты.

4.1 Программное обеспечение и Оборудование

В качестве программной библиотеки для реализации нейронных сетей был выбран Tensorflow [13]. Он имеет большой функционал и поддерживает обучение на видеопроцессорах (GPU). Немаловажной задачей является оптимальный способ подачи данных во время обучения, так как обучающая выборка имеет размер больше, чем доступное количество оперативной памяти на многих серверах (размер выборки $>$ 500 Гб). Для этого используется `td.data API`, который предоставляет удобный интерфейс для определения обучающей выборки, разделения ее на пакеты обучения, добавления стадии предобработки данных и случайного перемешивания примеров обучения.

Для обучения был использован сервер с 4-мя видеокартами Nvidia Tesla V100. Процесс подачи пакетов обучения построен так, что $n+1$-ый пакет подготавливается на центральных процессорах (ядрах CPU) в то время, пока n-ый пакет обучения обрабатывается на GPU. С применением техник оптимизации, которые доступны в Tensorflow, процент утилизации GPU во время обучения достигает примерно 99%. Визуальное представление этого процесса показано на рисунке 15.

Рис. 15: Визуальное представление процесса обучения на GPU [21].

4.2 Многослойный перцептрон

4.2.1 Входные данные

Входные данные, используемые для обучения многослойного перцептрона, состоят из 5 значений для каждого события в детекторе:

- Полное количество захваченных всеми ФЭУ фотонов ($N_{p.e.}$);
- Среднее значение времени захвата первого фотона, вычисленное по всем ФЭУ;
- Три координаты позиции вершины взаимодействия, полученные с помощью метода среднего заряда (см. раздел 1.1).

Обучающая выборка состоит из 140 000 событий, тестовая выборка состоит из 16 000 событий. Энергия позитрона во всех событиях составляет 5 МэВ. Так как эксперимент JUNO будет запущен только приблизительно в середине 2021 года, события для выборок были получены с помощью симуляции методом Монте-Карло. Симуляция была произведена с помощью официального фреймворка JUNO [22], который описывает физические и оптические процессы, происходящие внутри центрального детектора.

4.2.2 Архитектура

Архитектура использованного многослойного перцептрона преимущественно состоит из входного, выходного и скрытых слоев. Визуализация структуры нейронной сети показана на рис. 16.

Гиперпараметры, такие как количество нейронов в скрытых слоях, количество скрытых слоев, выбор метода минимизации функции потерь и другие, варьировались для того, чтобы найти оптимальную архитектуру. Весь список гиперпараметров, которые были подвержены поиску, перечислены в таблице 1. Поиск гиперпараметров производился с помощью перебора по сетке (Grid Search). Такой метод является достаточно эффективным ввиду того, что многослойный перцептрон не требует большого количества времени для обучения на современных видеопроцессорах.

Рис. 16: Архитектура использованного многослойного перцептрона после определения гиперпараметров с помощью перебора по сетке (Grid Search).

Гиперпараметр	Диапазон значений
Кол-во нейронов в скрытых слоях	128 - 8192
Кол-во скрытых слоев	2 - 10
Функция активации	ReLU / гиперболический тангенс
Кол-во событий в одном пакете обучения	16 - 8192
Метод оптимизации	Градиентный спуск / Adam
Начальная скорость обучения	0.00005 - 0.1
Коэффициент подавления скорости обучения	0.90 - 0.999
Коэффициент для слоя прореживания (dropout)	0.01 - 0.99
Нормализация входных данных	да / нет

Таблица 1: Список гиперпараметров и опций архитектуры сети.

Таким образом, архитектура использованной модели и процесс обучения после перебора гиперпараметров по сетке определяются следующим образом:

- Функция активации для всех нейронов скрытых слоев: блок линейной ректификации (ReLU);
- Функция активации для выходного слоя: отсутствует;
- Метод инициализации весовых коэффициентов: *нормализованная инициализация Глоро* [23];
- Функция потерь: среднеквадратическая ошибка (MSE);
- Метод минимизации функции потерь: Adam [24];
- Начальная скорость обучения равна 0.0003;
- Пакет обучения состоит из 8192 событий, обучение происходит на протяжении 100 эпох;
- Обучающие примеры случайно перемешиваются.

Было практически проверено, что Евклидово расстояние, использованное в качестве функции потерь, не вносит влияния на точность обучения, однако является немного более вычислительноемким при обучении сети на видеопроцессорах (GPU), что увеличивает время тренировки, хоть и незначительно.

В качестве метода оптимизации был выбран Adam [24]. Этот алгоритм вычислительно эффективен, имеет низкие требования к ресурсам памяти, и на практике показывает хорошую сходимость в задачах с большими объемами данных и/или параметров. Правило обновления параметров θ на каждом шаге обучения описано в главе 2 в публикации [24]:

$$
\begin{aligned}
&t = t + 1, \\
&\mathrm{lr}_t = \alpha \cdot \sqrt{\frac{1 - \beta_2^t}{1 - \beta_1^2}}, \\
&m_t = \beta_1 \cdot m_{t-1} + (1 - \beta_1) \cdot g, \\
&v_t = \beta_2 \cdot v_{t-1} + (1 - \beta_2) \cdot g \cdot g, \\
&\theta_t = \theta_{t-1} - \mathrm{lr}_t \cdot \frac{m_t}{\sqrt{v_t} + \epsilon},
\end{aligned}
$$

где t — это номер текущей итерации обучения, lr_t — это скорость обучения на итерации t, α — это начальная скорость обучения, ϵ — некоторая константа для предотвращения деления на ноль, β_1 и β_2 — константы из интервала $(0; 1)$ со значением близким к 1. Были использованы стандартные коэффициенты для метода Adam, которые рекомендованы авторами публикации [24]: $\beta_1 = 0.9$, $\beta_2 = 0.999$ и $\epsilon = 10^{-8}$. В качестве начальной скорости обучения α значение $3 \cdot 10^{-4}$ было найдено наиболее оптимальным.

4.2.3 Результаты

Результаты тестирования показаны на рис. 17. Зависимость между истинным и предсказанным радиусами, которая присутствует в методе среднего заряда, намного менее выражена в случае многослойного перцептрона. Необходимо отметить, что результаты содержат выбросы со значением $\Delta R > 40$ сантиметров, которые происходят из-за наличия выбросов в результатах метода среднего заряда, и не могут быть скорректированы нейронной сетью. На рисунке видно, что точность определения углов возврастает при возрастании радиуса события.

Стандартное отклонение в определении радиуса составляет примерно 16 сантиметров. Стандартное отклонение углов θ и ϕ равны примерно 0.02 и 0.05 радиан соответственно. При этом, использованный многослойный перцептрон не требует большого количества вычислительных ресурсов для обучения. Процесс обучения на видеокарте Nvidia Tesla V100 занимает примерно 2 минуты. Более сложные архитектуры моделей нейронных

сетей и их результаты будут рассмотрены в следующих главах. Результаты тестирования также нарисованы в зависимости от истинных значений координат и показаны на рис. 26 в приложении.

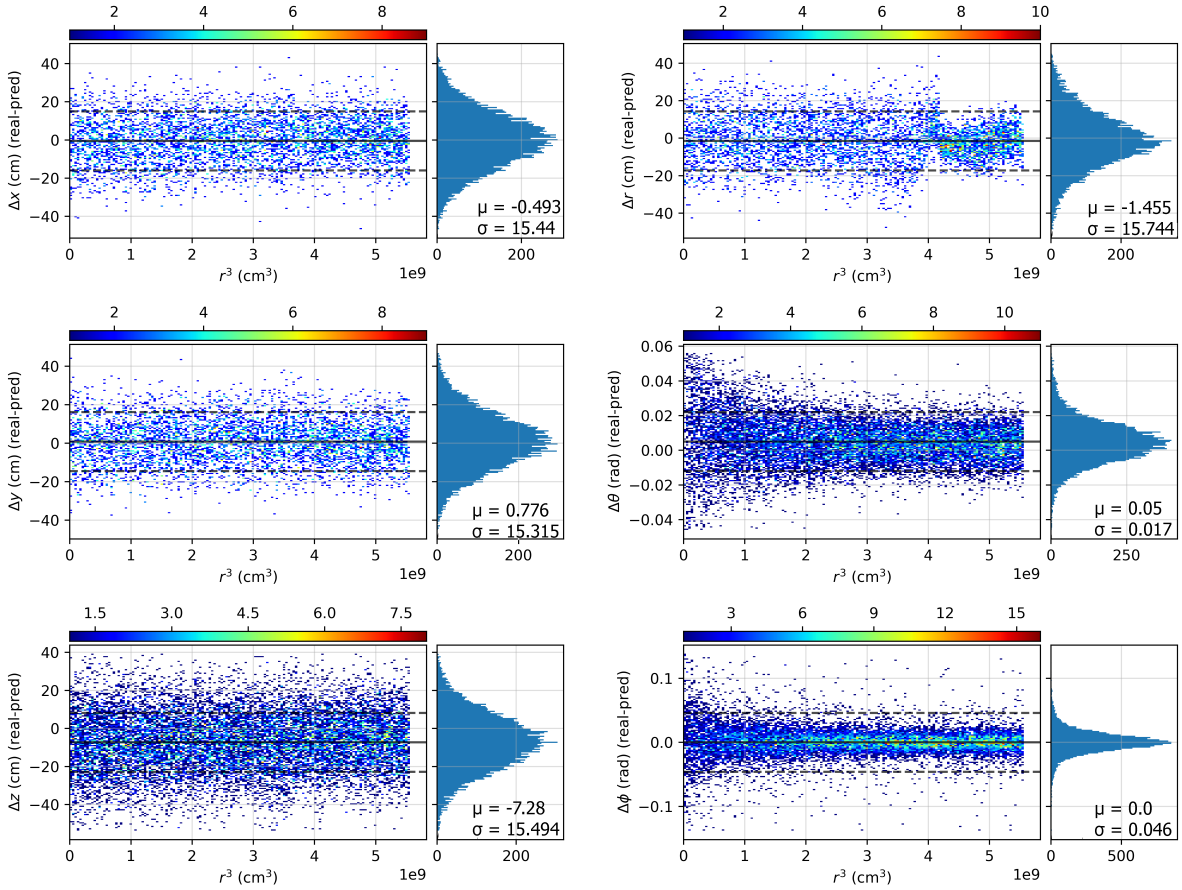


Рис. 17: Результаты тестирования многослойного перцептрона на тестовой выборке из 16 000 событий с энергией 5 МэВ. На рисунке изображена разница в евклидовых и сферических координатах между реконструированной и истинной позициями вершины в зависимости от радиуса в кубе.

4.3 Сверточная нейронная сеть

4.3.1 Входные данные

Входными данными для каждого события являются 2-канальные изображения разрешением 256x128 пикселей (см. рис. 18). Первый канал изображения содержит информацию о принятом заряде всех ФЭУ, а второй — о времени прихода каждого фотона. Позиции всех ≈ 20000 больших ФЭУ были спроектированы на плоскость с помощью проекции Мольвейде (см.

3). Дополнительная нормализация входных данных не применялась, согласно факту, что при использовании активации ReLU это не обязательно [14].

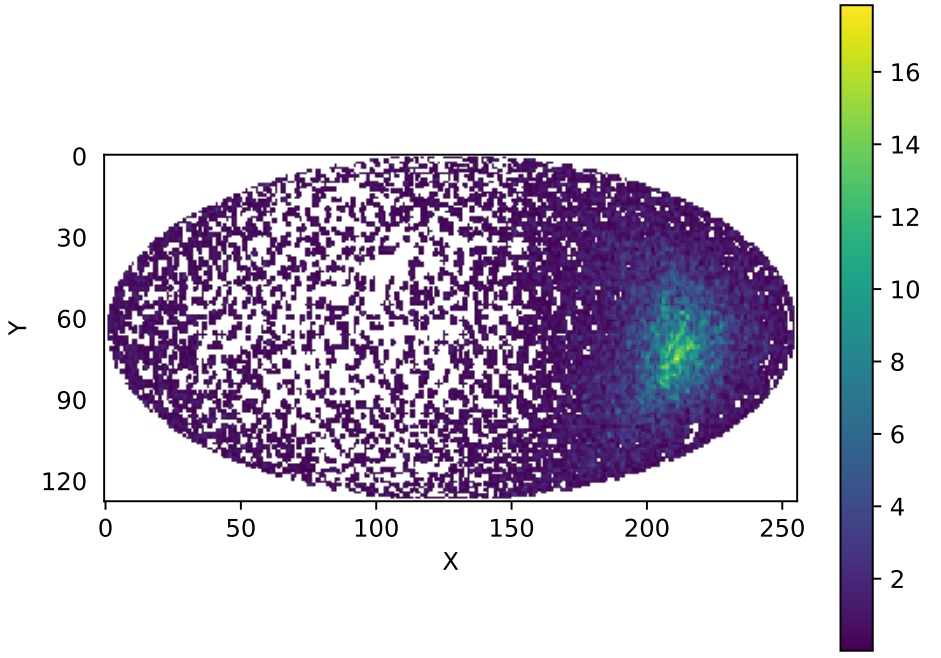
Обучающая выборка состоит из 140 000 событий с энергией 5 МэВ. Тестовая выборка состоит из 16 000 событий с аналогичной энергией. Истинные позиции вершин взаимодействия равномерно распределены внутри детектора.

4.3.2 Архитектура

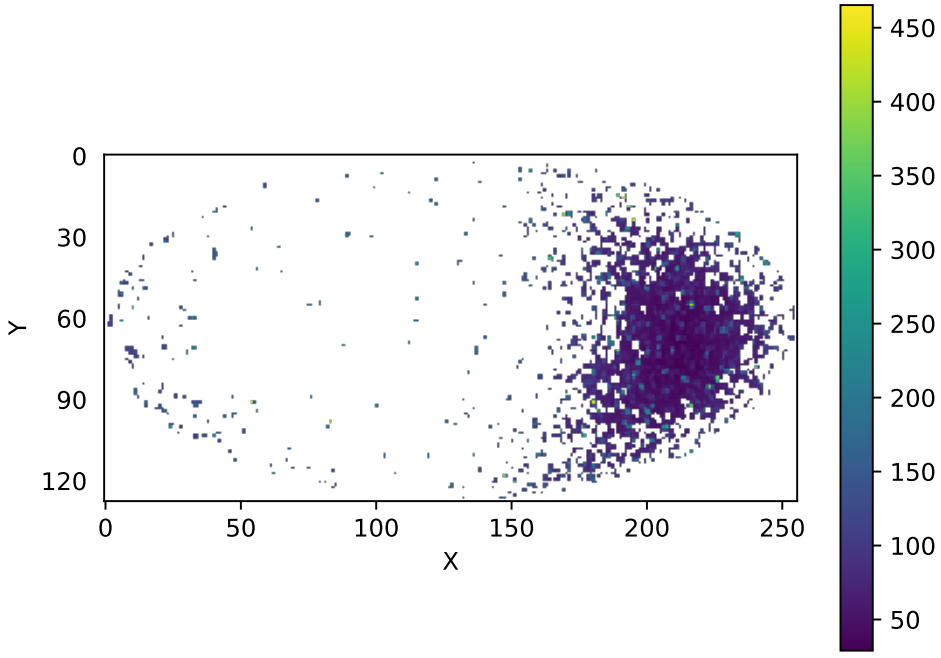
Сеть имеет 1 входной слой, 4 слоя свертки, 4 слоя пулинга, 2 полносвязных слоя и 1 выходной слой. После каждого слоя свертки дополнительно применяется слой локальной нормализации сигнала (local response normalization) [25], чтобы улучшить регуляризующую способность сети. Слой прореживания применяется перед выходным слоем, чтобы снизить эффект переобучения. Визуализация архитектуры используемой сети представлена на рис. 19.

Таким образом, параметры обучения выглядят следующим образом:

- Функция активации для полносвязных слоев и слоев свертки, кроме последнего полносвязного слоя: ReLU. Последний полносвязный слой не имеет функции активации;
- Метод инициализации весов: нормальная инициализация Глоро [23];
- L2 Регуляризация весов с коэффициентами регуляризации 0.01 и 0.05 для сверточных и полносвязных слоев соответственно;
- Коэффициенты для слоев локальной нормализации сигнала: $\alpha = 0.001/9.0$, $\beta = 0.75$, $k = 0.5$;
- Функция потерь: среднеквадратическая ошибка (MSE);
- Метод минимизации функции потерь: Adam;
- Начальная скорость обучения $3 \cdot 10^{-4}$;



(a)



(b)

Рис. 18: 2D изображения одного события, полученные после применения проекции Мольвейде. (a) — информация, соответствующая первому каналу (информации о заряде), (b) — соответствующая второму (информация о времени прихода фотонов). Ось Y соответствует углу θ, ось X углу ϕ.

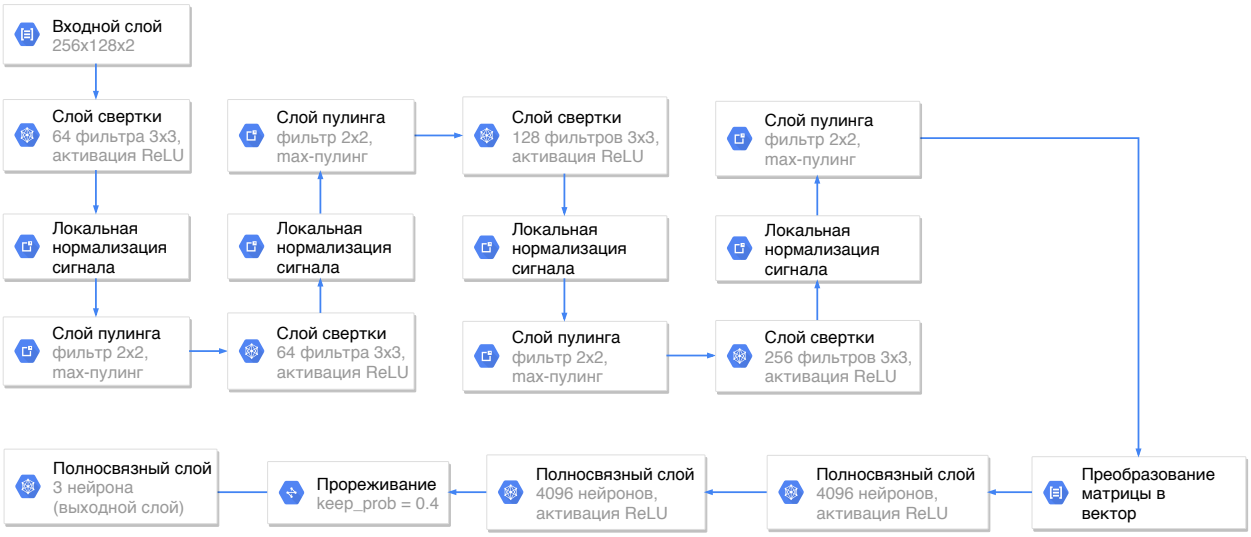


Рис. 19: Визуальное представление архитектуры используемой сверточной сети.

- Размер пакета 128 событий, обучение на протяжении 64 эпох;
- Обучающие примеры случайно перемешиваются.

4.3.3 Результаты

Результаты тестирования нейронной сети представлены на рис. 20. Тестовая выборка состоит из 16 000 событий с энергией 5 МэВ. Разница в определении истинного и реконструированного радиусов вершин взаимодействия имеет систематическое смещение в ≈ -1.35 сантиметра и стандартное отклонение ≈ 14.65 сантиметра. Результаты не показывают сильного ухудшения в точности определения позиции вершины при возрастании истинного радиуса события, как это происходит при использовании метода среднего заряда, однако также присутствуют события-выбросы с высоким значением разницы между истинным и реконструированным радиусами. В процессе исследования этой проблемы было выявлено, что больший размер обучающей выборки помогает уменьшить количество выбросов. Результаты тестирования также нарисованы в зависимости от истинных значений координат и показаны на рис. 27 в приложении.

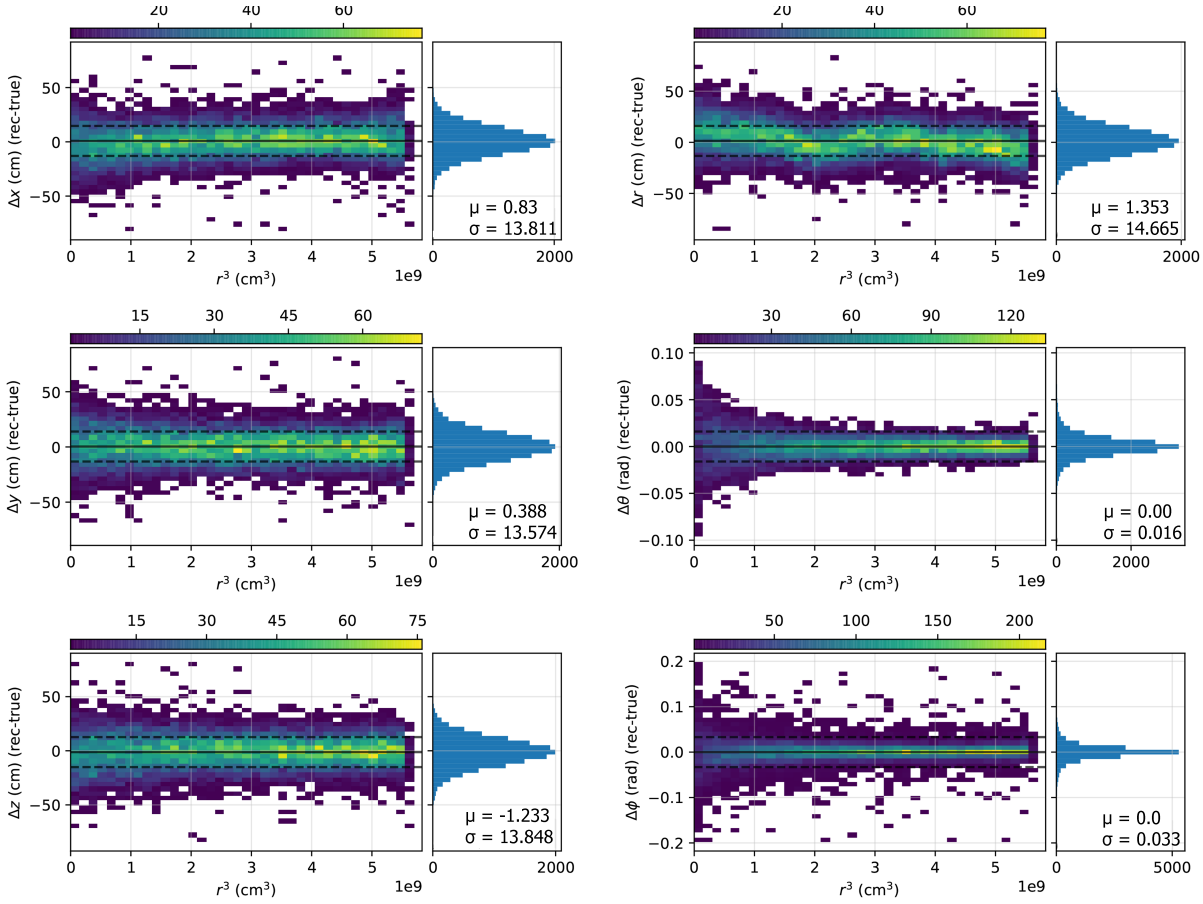
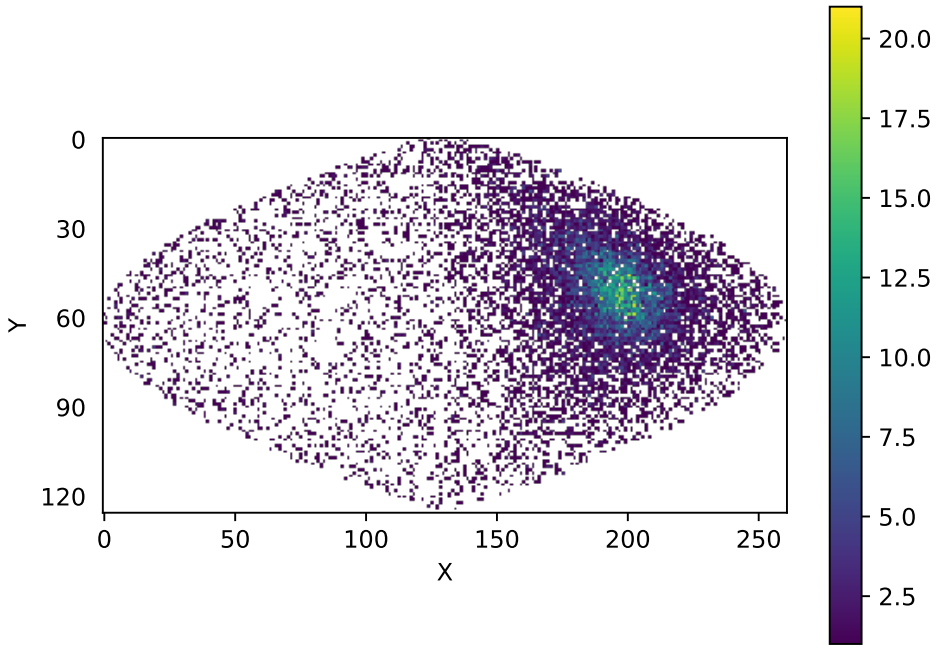


Рис. 20: Результаты реконструкции позиции вершины взаимодействия с помощью сверточной нейронной сети. На рисунке изображена разница в евклидовых и сферических координатах между реконструированными и истинными вершинами в зависимости от истинного радиуса в кубе. Цвет точки означает количество событий с данным значением. Сплошная черная горизонтальная линия обозначает систематическое смещение (μ), а верхние и нижние пунктирные линии значения $\mu + \sigma$ and μ - σ соответственно.

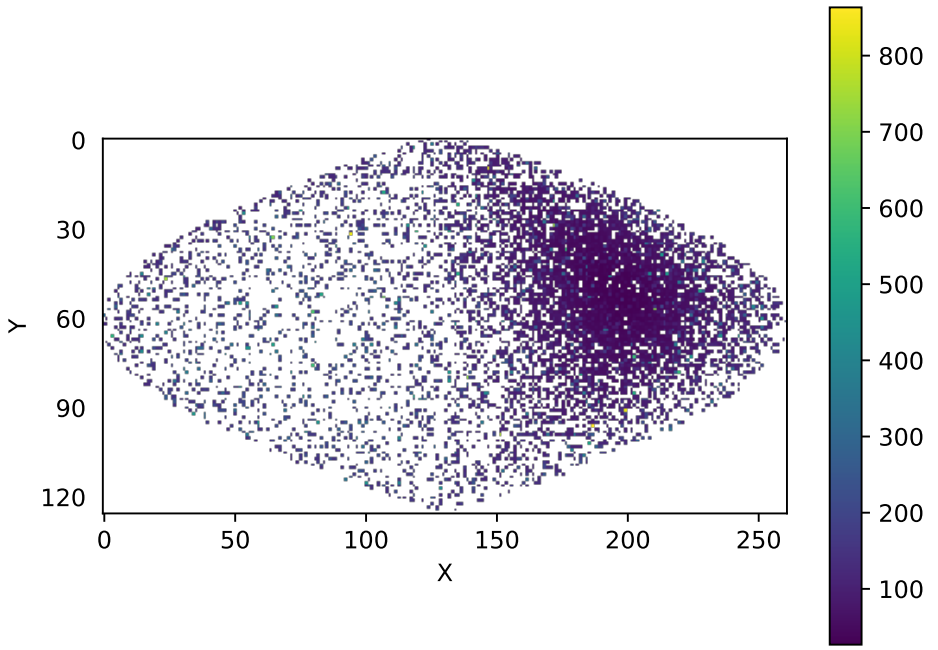
4.4 Остаточная сверточная нейронная сеть

4.4.1 Входные данные

Входными данными для каждого события являются 2-канальные изображения разрешением 256x128 пикселей (см. рис. 21). Каналы изображения идентичны входным данным из предыдущей главы — первый канал соответствует информации о заряде каждого ФЭУ, второй канал — о времени прихода фотонов. Отличие заключается в том, что вместо проекции Мольвейде используется синусоидальная проекция. Эмпирически было получено, что данная проекция лучше подходит для задачи реконструкции вершины взаимодействия в детекторе JUNO.



(a)



(b)

Рис. 21: 2D изображения одного события, полученные после применения синусоидальной проекции. (a) — информация, соответствующая первому каналу (информации о заряде), (b) — соответствующая второму (информация о времени прихода фотонов). Ось Y соответствует углу θ, ось X углу ϕ.

Размеры выборок были увеличены по сравнению с подходом в прошлой главе. Теперь обучающая выборка состоит из 900 000 событий. Собы-

тия равномерно распределены по истинной позиции вершины в детекторе, а также равномерны распределены по энергиям в интервале $[0, 10]$ МэВ. Тестовая выборка состоит из 100 000 событий с аналогичными свойствами.

4.4.2 Архитектура

В качестве основы сети была использована архитектура *ResNet-50* [9]. Данная сеть была модифицирована для использования в задаче реконструкции вершины взаимодействия. Гиперпараметры нейронной сети были оптимизированы с помощью инструмента Microsoft Neural Network Intelligence [26]. Был использован алгоритм Tree Parzen Estimator [27], доступный в этом инструменте. Этот метод эффективен для нейронных сетей, имеющих большую вычислительную сложность. Это актуально для данной остаточной сверточной сети, так как один полный процесс обучения занимает примерно 22 часа, и количество запусков сети за небольшой промежуток времени ограничено. После оптимизации гиперпараметров параметры обучения выглядят следующим образом:

- Архитектура сети состоит из 4 групп, в каждой из которой по 3, 4, 6 и 3 остаточных блока соответственно. Каждый остаточный блок содержит по 3 слоя свертки. Полное количество слоев свертки равно 49. Визуализация архитектуры сети представлена на рис. 22;
- Функция активации: ReLU с утечкой (Leaky ReLU) [28]. У последнего полносвязного слоя функция активации отсутствует;
- Метод инициализации весов: нормализованная инициализация Xe [29];
- L2 регуляризация весов с коэффициентом регуляризации 0.0001;
- Коэффициенты для пакетной нормировки: momentum $= 0.9$, epsilon $= 1e - 5$;
- Функция потерь: среднеквадратическая ошибка (MSE);
- Метод минимизации функции потерь: Adam;

- Начальная скорость обучения равна 0.001. Скорость обучения умножается на коэффициент 0.95 каждые 10000 шагов обучения;
- Размер пакета обучения равен 64 событиям, обучение производится на протяжении 100 эпох;
- Обучающие примеры случайно перемешиваются.

4.4.3 Результаты

Тестирование нейронной сети было произведено на выборке из 100 000 событий. Позиции истинных вершин событий в выборке равномерно распределены внутри детектора. Энергии событий равномерно распределены в интервале $[0, 10]$ МэВ. На рисунке 23 показано, что разница между истинными и реконструированными радиусами вершин взаимодействия имеет систематическое смещение ≈ -0.016 сантиметра и стандартное отклонение ≈ 4.93 сантиметра. На рисунке 24 изображены значения систематического смещения и стандартного отклонения в зависимости от истинного радиуса события. На этом изображении видно, что метод способен реконструировать вершины событий, близких к краю детектора, без значительного увеличения стандартного отклонения. Значения стандартного отклонения для событий с фиксированной энергией {1, 2, 3, ... 10} МэВ представлены на рисунке 25. Для тестирования использовалось 2000 событий для каждого значения энергии. Кроме этого, разница в евклидовых и сферических координатах между реконструированными и истинными вершинами показана на рисунках 28, 29 в приложении.

Рис. 22: Визуальное представление архитектуры использованной остаточной сверточной сети.

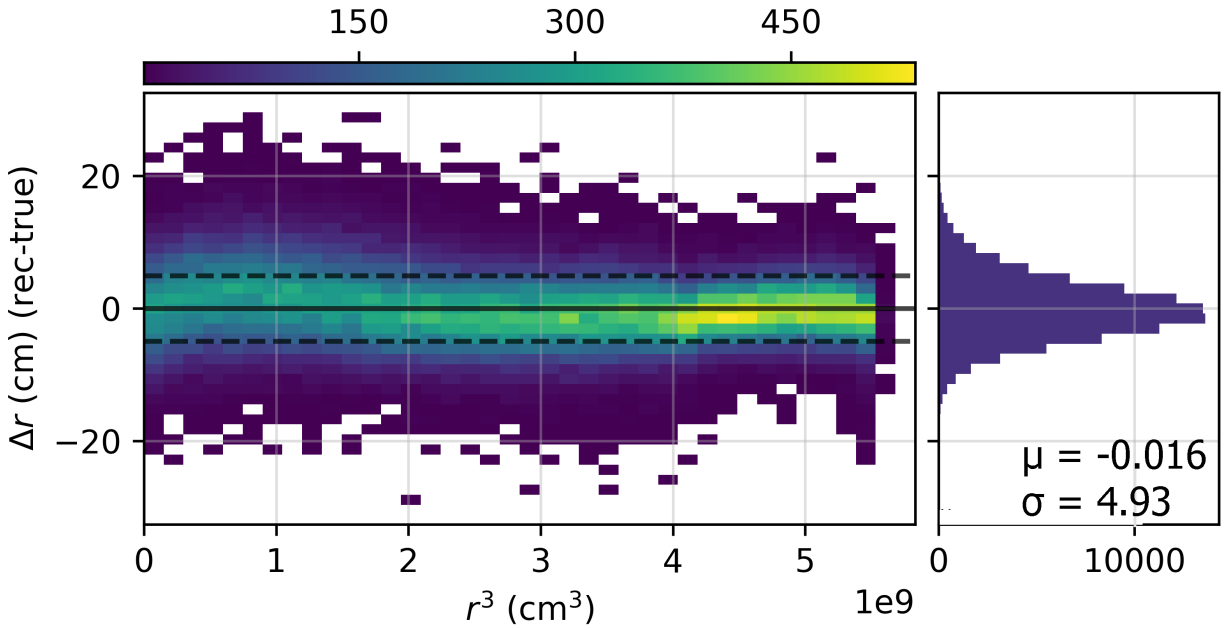


Рис. 23: Разница между реконструированным и истинным радиусами вершины взаимодействия в зависимости от истинного радиуса в кубе. Цвет точки обозначает количество событий с данным значением. Тестовая выборка состоит из 100 000 событий с энергиями $[0-10]$ МэВ. Сплошная черная горизонтальная линия обозначает систематическое смещение (μ), а верхние и нижние пунктирные линии значения $\mu + \sigma$ and μ - σ соответственно.

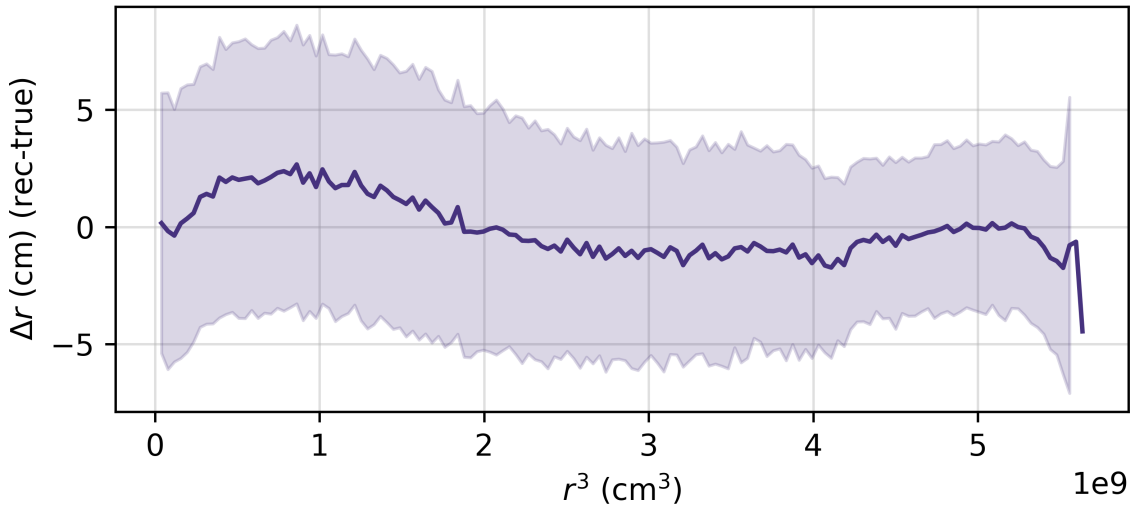


Рис. 24: График зависимости систематического смещения (μ) и стандартного отклонения (σ) разницы между реконструированными и истинными радиусами вершин взаимодействия в зависимости от истинного радиуса в кубе. Тестовая выборка состоит из 100 000 событий с энергиями $[0-10]$ МэВ. Линия обозначает значение систематического смещения, а ширина полупрозрачной полосы — стандартное отклонение.

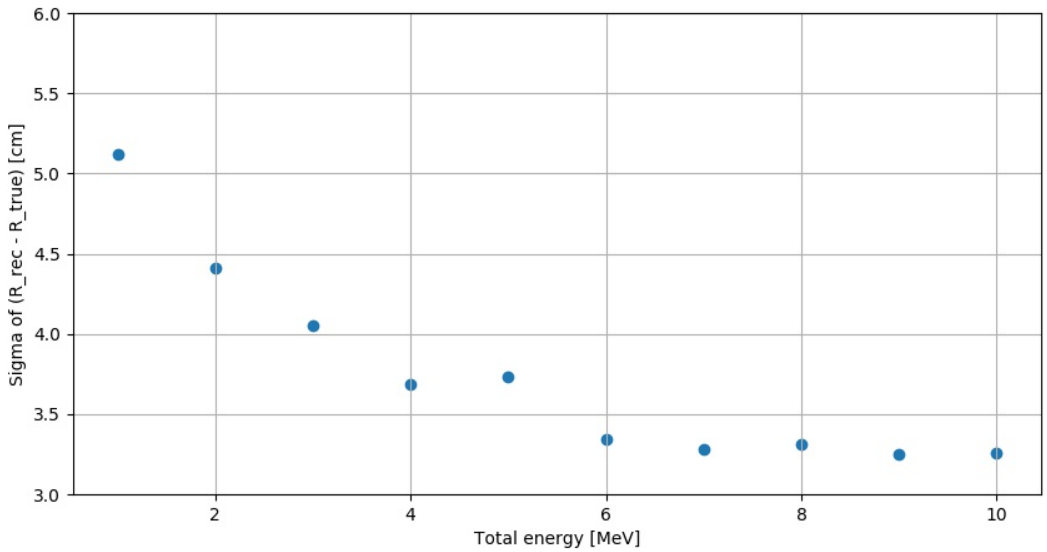


Рис. 25: Значение стандартного отклонения разницы между реконструированными и истинными радиусами вершин взаимодействия для событий с дискретными энергиями 1, 2, ... , 10 МэВ. Тестовая выборка состоит из 2000 событий для каждого значения энергии.

Выводы

Из результатов, описанных в главе 4 и таблицах 2, 3 можно сделать следующие выводы:

- Поставленная задача была решена полностью;
- Нейронные сети показывают сопоставимые с классическими методами результаты в терминах точности реконструкции вершины взаимодействия;
- Остаточная сверточная нейронная сеть, использованная в этой работе, реконструирует вершину взаимодействия в JUNO точнее, чем использованные многослойный перцептрон и сверточная нейронная сеть;
- Нейронные сети могут использоваться как инструмент для анализа больших объемов данных в JUNO, так как время на обработку одного события после успешного обучения сети относительно мало.

Название метода	Стандартное отклонение величины Δr (см)	Систематическое смещение величины Δr (см)	Энергия событий в тестировании (МэВ)
Метод среднего заряда	132.82	291.36	5
Многослойный перцептрон	15.744	-1.455	5
Сверточная нейронная сеть	14.665	1.353	5
Метод максимального правдоподобия	≈ 7	< 3	1
Остаточная сверточная нейронная сеть	5.10	0.77	1
Остаточная сверточная нейронная сеть	4.93	-0.016	$0 - 10$

Таблица 2: Таблица стандартных отклонений и систематических смещений разницы между реконструированными и истинными радиусами вершин взаимодействия в центральном детекторе JUNO для описанных в этой работе методов.

Название метода	Время на реконструкцию вершины одного события	Энергия событий в тестировании (МэВ)
Метод максимального правдоподобия (CPU)	$\approx$ 14.19 сек.	10
Метод максимального правдоподобия (CPU)	$\approx$ 1.88 сек.	1
Сверточная нейронная сеть (GPU)	$\approx$ 100 миллисекунд	5
Метод максимального правдоподобия (GPU)	$\approx$ 95 миллисекунд	10
Метод максимального правдоподобия (GPU)	$\approx$ 50 миллисекунд	1
Остаточная сверточная нейронная сеть (GPU)	$\approx$ 1.2 миллисекунды	0 — 10

Таблица 3: Таблица значений времени на реконструкцию вершины одного события в зависимости от метода. Для нейронных сетей время предсказания указано после стадии обучения и для запуска на видеокарте Nvidia Tesla V100, 20 ядрах CPU и 100Гб оперативной памяти. Для остаточной сверточной нейронной сети время реконструкции не зависит от энергии события.

Заключение

Целью этой работы было разработать метод для реконструкции первичной вершины взаимодействия в центральном детекторе JUNO с использованием нейронных сетей. Три различные архитектуры сетей были реализованы и протестированы. Обзорная информация по численным характеристикам точности и временным затратам на реконструкцию представлены в таблицах 2 и 3. Остаточная сверточная нейронная сеть решает поставленную задачу, полностью соответствуя требованиям. Проведенное исследование показывает, что нейронные сети могут применяться для анализа большого объема данных в эксперименте JUNO.

Благодарности

Прежде всего я хочу поблагодарить Константина Трескова, Олега Якушкина и Валерия Гришкина за научное руководство. Большое спасибо Максиму Гончару за многочисленные консультации и поддержку в исследовании. Также я хочу поблагодарить Дмитрия Наумова за возможность работать в Лаборатории ядерных проблем Объединенного института ядерных исследований.

Большое спасибо Лаборатории информационных технологий ОИЯИ

и команде платформы Hybrilit, а в особенности Дмитрию Подгайному, Оксане Стрельцовой, Дмитрию Белякову, Максиму Зуеву и Михаилу Матвееву за предоставление доступа к вычислительным ресурсам [30]. Эти вычислительные ресурсы включают в себя сервер с 4-мя видеокартами Nvidia Tesla V100 и необходимым количеством ОЗУ и ядер CPU. Я благодарю Николая Кутовского за предоставление доступа к облачным ресурсам ЛИТ ОИЯИ [31], которые включают виртуальный сервер с видеокартой Nvidia Tesla V100. Кроме этого, я хочу поблагодарить Анну Фатькину и Ивана Ганкевича за предоставление доступа к ресурсам на серверах СПбГУ.

Список литературы

[1] Fengpeng An et al. Neutrino physics with JUNO. Journal of Physics G: Nuclear and Particle Physics, 43(3):030401, Mar 2016.

[2] Gioacchino Ranucci, JUNO Collaboration, et al. Status and prospects of the juno experiment. In Journal of Physics: Conference Series, volume 888, page 012022. IOP Publishing, 2017.

[3] Cheng Yaping et al. GNA fitter and Detector response impact on MH sensitivity study, June 2018. https://doi.org/10.5281/zenodo.1289188 (дата посещения: 20 Мая 2019).

[4] Q. Liu, M. He, X. Ding, W. Li, and H. Peng. A vertex reconstruction algorithm in the central detector of JUNO. ArXiv e-prints, March 2018.

[5] Jiang Zhu, Zhengyun You, Yumei Zhang, Ziyuan Li, Shu Zhang, Tao Lin, and Weidong Li. A method of detector and event visualization with unity in juno. Journal of Instrumentation, 14(01):T01007, 2019.

[6] Timothée Brugière. The Jiangmen underground neutrino observatory experiment. Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment, 845:326–329, 2017.

[7] T. Adam, F. An, G. An, Q. An, N. Anfimov, V. Antonelli, G. Baccolo, M. Baldoncini, E. Baussan, M. Bellato, and et al. JUNO Conceptual Design Report. ArXiv e-prints, August 2015.

[8] Ян Гудфеллоу, Бенджио Иошуа, and Аарон Курвилль. Глубокое обучение. Litres, 2018.

[9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 770–778, 2016.

[10] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. Journal of Machine Learning Research, 15:1929–1958, 2014.

[11] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. arXiv preprint arXiv:1502.03167, 2015.

[12] John P. Snyder. Map projections - a working manual. Number ARRAY(0x484ba10) in Professional paper / US Department of the Interior, US Geological Survey. United States Gov. Print. Office, Washington, DC [u.a.], 1987. Supersedes USGS bulletin 1532.

[13] Martín Abadi et al. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.

[14] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, Advances in Neural Information Processing Systems 25, pages 1097–1105. Curran Associates, Inc., 2012.

[15] Y-T Zhou, Rama Chellappa, Aseem Vaid, and B Keith Jenkins. Image restoration using a neural network. IEEE Transactions on Acoustics, Speech, and Signal Processing, 36(7):1141–1151, 1988.

[16] Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. Understanding the exploding gradient problem. CoRR, abs/1211.5063, 2, 2012.

[17] Sepp Hochreiter. The vanishing gradient problem during learning recurrent neural nets and problem solutions. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, 6(02):107–116, 1998.

[18] Wikipedia contributors. List of map projections — Wikipedia, the free encyclopedia, 2018. `https://en.wikipedia.org/wiki/List_of_map_projections` (дата посещения: 20 Мая 2019).

[19] Wikipedia contributors. Mollweide projection — Wikipedia, the free encyclopedia, 2019. https://en.wikipedia.org/wiki/Mollweide_projection (дата посещения: 20 Мая 2019).

[20] Wikipedia contributors. Sinusoidal projection — Wikipedia, the free encyclopedia, 2019. https://en.wikipedia.org/wiki/Sinusoidal_projection (дата посещения: 20 Мая 2019).

[21] Google. Data input pipeline performace | tensorflow core | tensorflow, 2019. https://www.tensorflow.org/guide/performance/datasets (дата посещения: 20 Мая 2019).

[22] Tao Lin, Jiaheng Zou, Weidong Li, Ziyan Deng, Xiao Fang, Guofu Cao, Xingtao Huang, Zhengyun You, et al. The application of sniper to the juno simulation. In Journal of Physics Conference Series, volume 898, 2017.

[23] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In Proceedings of the thirteenth international conference on artificial intelligence and statistics, pages 249–256, 2010.

[24] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. CoRR, abs/1412.6980, 2014.

[25] Alan E Robinson, Paul S Hammon, and Virginia R de Sa. Explaining brightness illusions using spatial filtering and local response normalization. Vision research, 47(12):1631–1644, 2007.

[26] Microsoft. An open source automl toolkit for neural network architecture search and hyper-parameter tuning, 2019. https://github.com/microsoft/nni (дата посещения: 20 Мая 2019).

[27] James S Bergstra, Rémi Bardenet, Yoshua Bengio, and Balázs Kégl. Algorithms for hyper-parameter optimization. In Advances in neural information processing systems, pages 2546–2554, 2011.

[28] Andrew L Maas, Awni Y Hannun, and Andrew Y Ng. Rectifier nonlinearities improve neural network acoustic models. In Proc. icml, volume 30, page 3, 2013.

[29] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In Proceedings of the IEEE international conference on computer vision, pages 1026–1034, 2015.

[30] E.I. Alexandrov, D.V. Belyakov, M.A. Matveyev, D.V. Podgainy, O.I. Streltsova, E.V. Zemlyanaya, P.V. Zrelov, M.I. Zuev, et al. Research of acceleration calculations in solving scientific problems on the heterogeneous cluster hybrilit. Вестник Российского университета дружбы народов. Серия: Математика, информатика, физика, (4), 2015.

[31] A.V. Baranov, N.A. Balashov, N.A. Kutovskiy, and R.N. Semenov. Jinr cloud infrastructure evolution. Physics of Particles and Nuclei Letters, 13(5):672–675, 2016.

Приложение

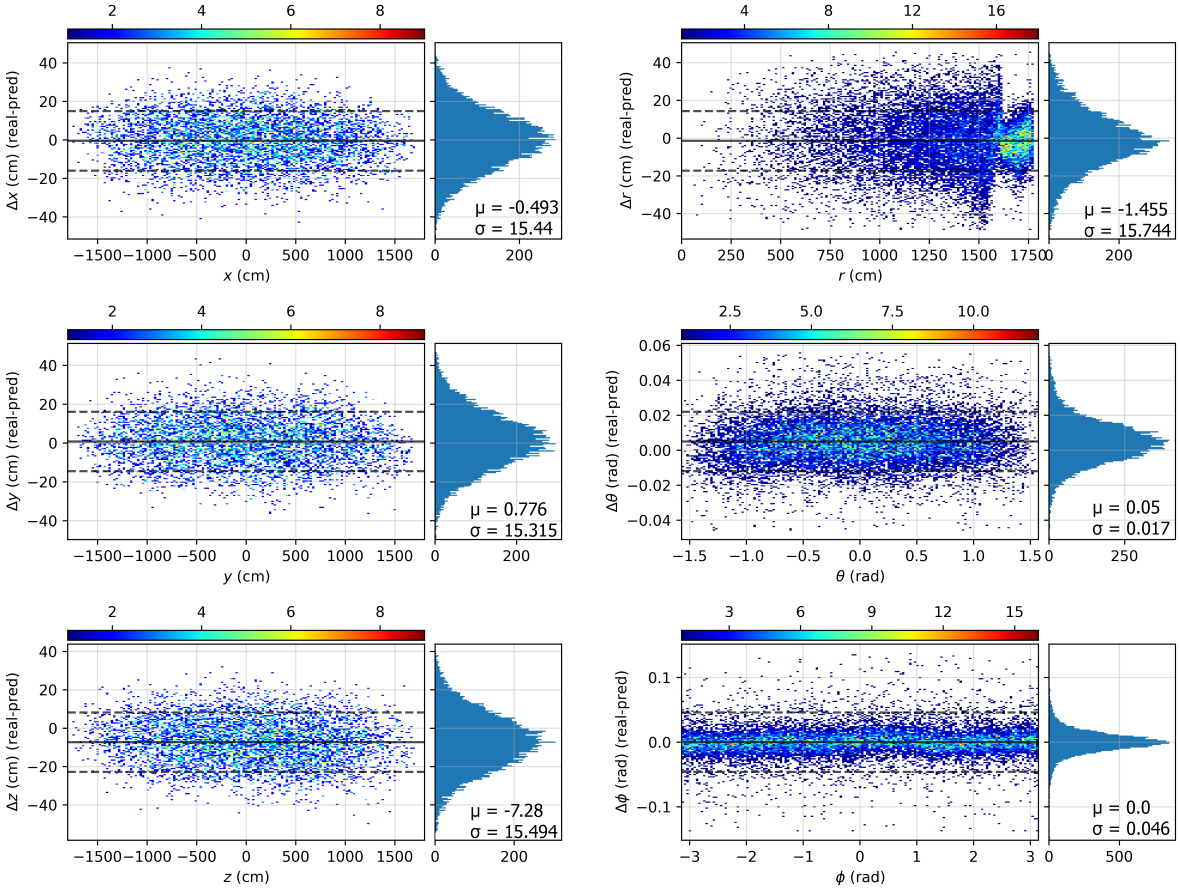


Рис. 26: Результаты тестирования многослойного перцептрона на тестовой выборке из 16 000 событий с энергией 5 МэВ. На рисунке изображена разница в евклидовых и сферических координатах между реконструированной и истинной позициями вершины.

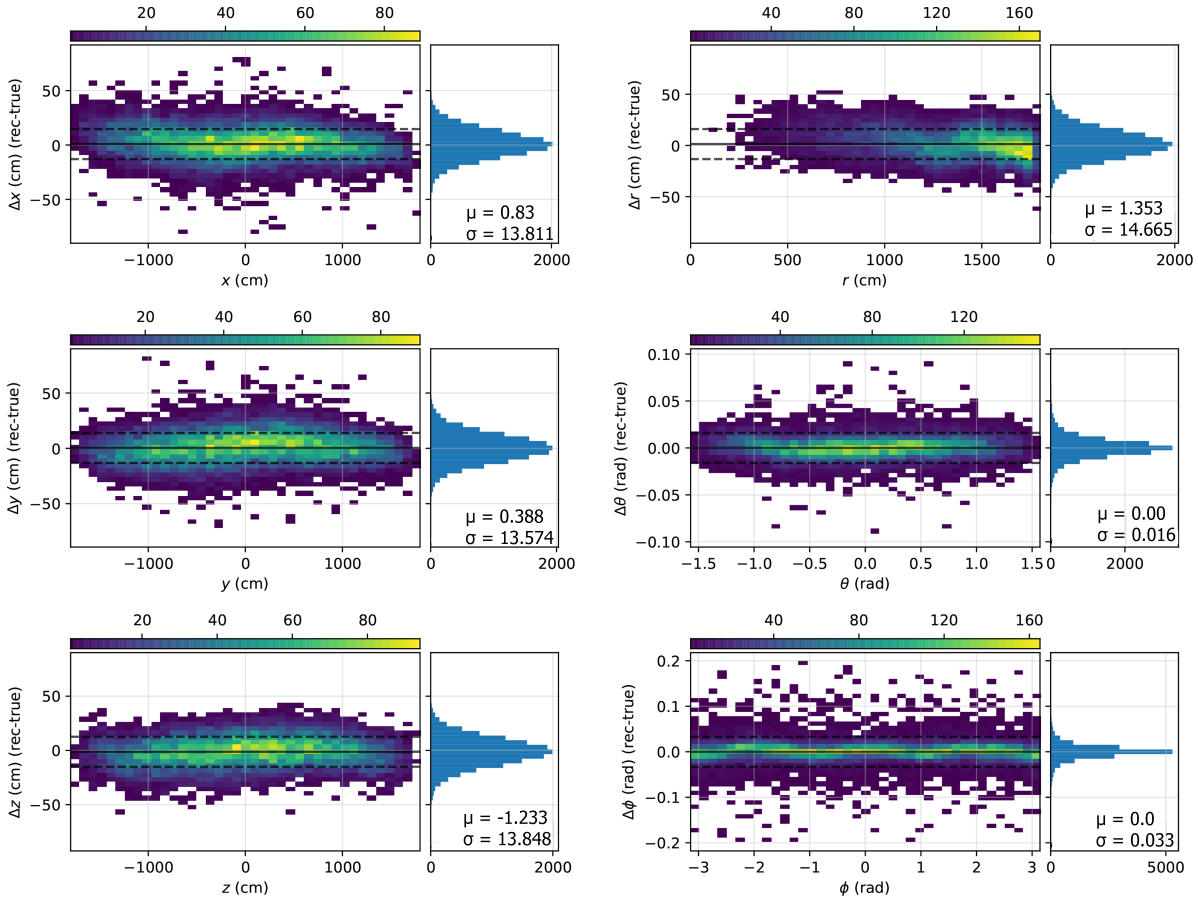


Рис. 27: Результаты реконструкции позиции вершины взаимодействия с помощью сверточной нейронной сети. На рисунке изображена разница в евклидовых и сферических координатах между реконструированными и истинными вершинами. Цвет точки означает количество событий с данным значением. Сплошная черная горизонтальная линия обозначает систематическое смещение (μ), а верхние и нижние пунктирные линии значения $\mu + \sigma$ and μ - σ соответственно.

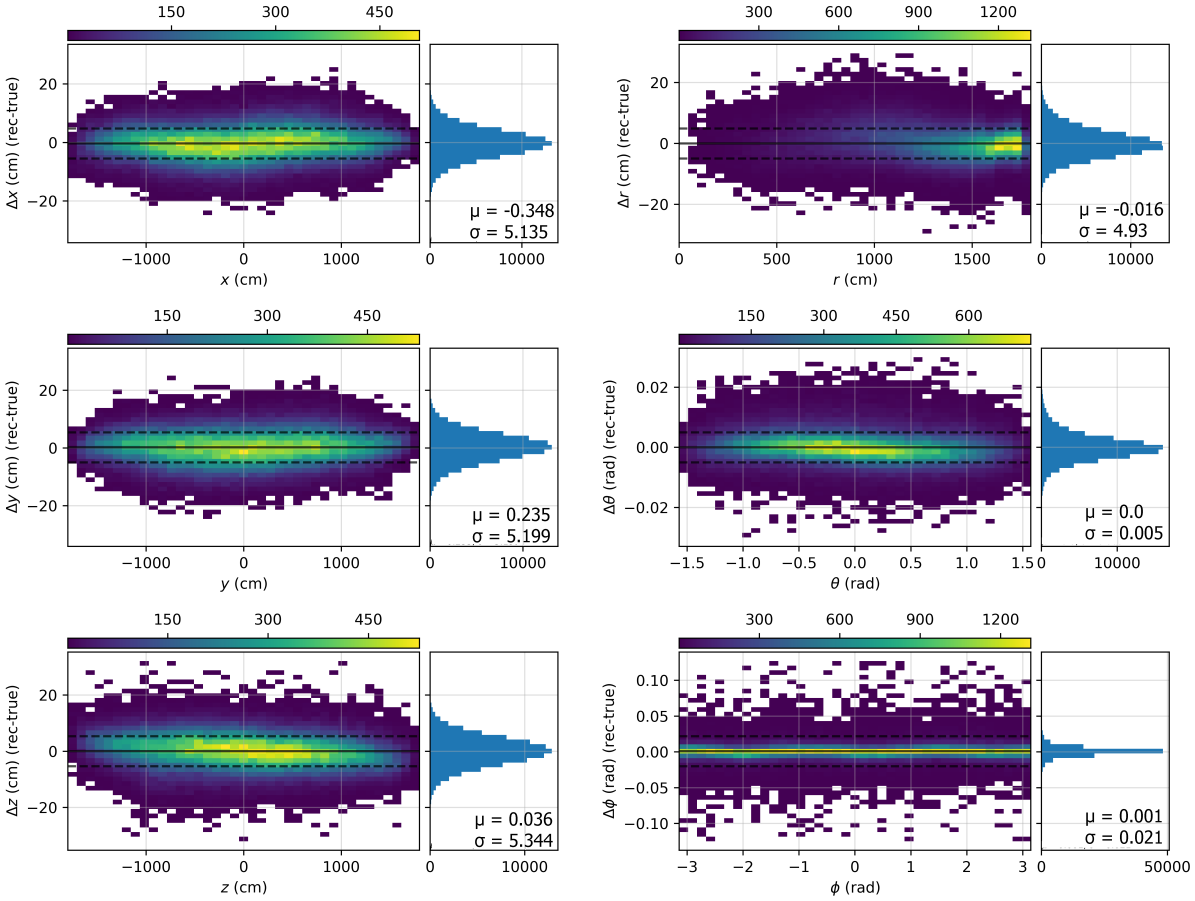


Рис. 28: Результаты реконструкции позиции вершины взаимодействия с помощью остаточной сверточной нейронной сети. Тестовая выборка состоит из 100 000 событий с энергиями $[0 - 10]$ МэВ. На рисунке изображена разница в евклидовых и сферических координатах между реконструированными и истинными вершинами. Цвет точки означает количество событий с данным значением. Сплошная черная горизонтальная линия обозначает систематическое смещение (μ), а верхние и нижние пунктирные линии значения $\mu + \sigma$ and μ - σ соответственно.

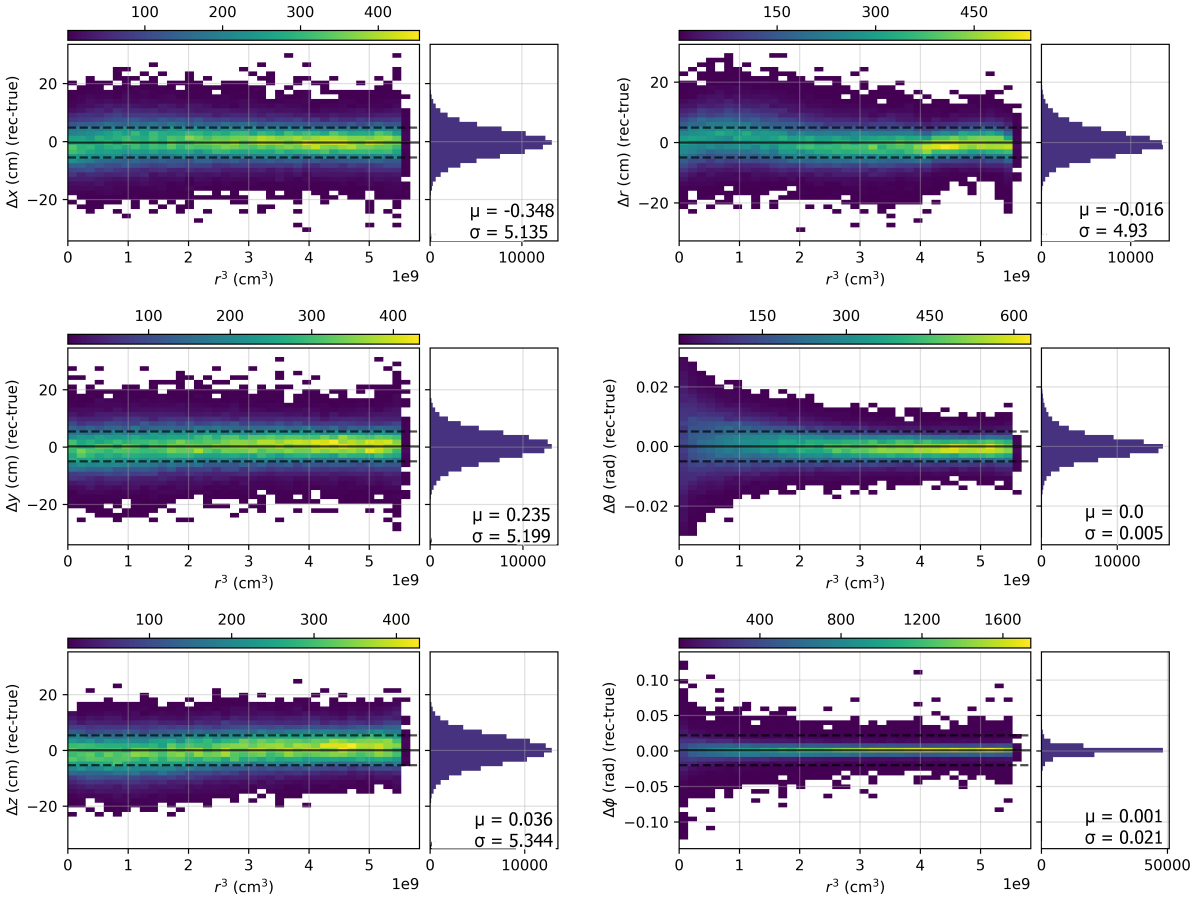


Рис. 29: Результаты реконструкции позиции вершины взаимодействия с помощью остаточной сверточной нейронной сети. Тестовая выборка состоит из 100 000 событий с энергиями $[0 - 10]$ МэВ. На рисунке изображена разница в евклидовых и сферических координатах между реконструированными и истинными вершинами в зависимости от истинного радиуса в кубе. Цвет точки означает количество событий с данным значением. Сплошная черная горизонтальная линия обозначает систематическое смещение (μ), а верхние и нижние пунктирные линии значения $\mu + \sigma$ and μ - σ соответственно.